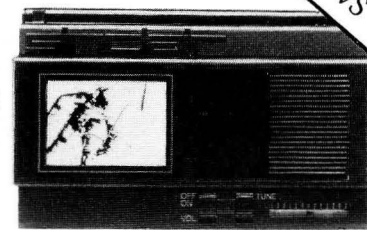


QLUB

News for the QL user



**EXTRA
INSIDE**
YOUR CHANCE TO
WIN ONE OF THREE
SINCLAIR POCKET TVS

ISSUE No. 4 MARCH/APRIL 1985

CARTRIDGE PRICES CUT

THE age of the £1.99 Microdrive cartridge is here. As Sinclair Research's Nigel Searle has promised, the cartridge prices were slashed when production reached large enough volumes to meet the demand for a better-priced cartridge.

THE price cut was announced in early February and promises to stimulate even further development of software for the QL and should bring down the prices of the software that is produced. The news of the price cut follows a series of recent events which have boosted use of the Microdrive storage technology in the UK.

Both ICL and British Telecom have announced that they will use Microdrive and QL technology in their desktop business computer systems, while sales of the Sinclair Spectrum and Spectrum + — both of which can use Microdrives with

the Interface 1 expansion unit — hit new levels at Christmas time.

And while Microdrive prices were coming down, new Microdrive versions of the QL Psion software were being shipped to QLUB members. By the time you read this, all QLUB members who wanted them should have received the software updates of QL Quill, Abacus, Archive and Easel that their membership provides.

Details of the Version 2 software can be found in the special software pull-out section in this edition of QLUB News.

QLUB extended to all owners

**ALL QL owners are being offered a year's
free QLUB membership.**

EXISTING QLUB members will not have to pay for their membership to be extended through to February 1986 and therefore will be offered QLUB discounts, Psion software support and the QLUB News newsletters for a full year from now. And all new QL owners who fill out their guarantee registration cards will automatically qualify for a year's QLUB membership (excluding future software updates) from the time they send in those cards.

The offer is being made to further improve QLUB by allowing the widest possible group of QL users to contribute to and benefit from the newsletter and to receive the benefits of the Psion software support and QLUB special discounts. All QLUB members are entitled to a 20 per cent discount on software sold through QLUB including QL Chess—QLUB price £14.95, QL Toolkit—QLUB price £19.95 and QL Cash Trader—with a QLUB price of £54.95.

9 Software special



- Cashing in with QL Cash Trader
- Accounting on your QL with Sage
- Version 2 Psion software
- Software developer's news
- Project planning and decision making
- QL Toolkit

SOFTWARE SPECIAL

TO celebrate the QL's first birthday, we present a special eight-page pull-out software supplement detailing the latest in software releases for the machine and giving detailed information on how packages work, what the prices are and where you can get them.

Information is included on programming tools such as QL Toolkit, accounting software such as Sagesoft's QL Integrated Accounts and cash management programs such as Accountancy Software's QL Cash Trader.

INSIDE QLUB NEWS

| | |
|----------------------------|----|
| QLetters | 2 |
| CST disk interface | 4 |
| QLibrary | 5 |
| Joystick jottings | 6 |
| Sounding off | 18 |
| Machine code windows | 22 |

sinclair

Welcome to new QLUB Members

AS you've probably noticed, this issue marks the first anniversary of both the release of the QL and the start-up of QLUB. It's also an exciting time for us at QLUB News.

AND as you'll have seen from the news story on the front page, QLUB membership has been extended to all QL purchasers who fill out their warranty cards, while existing QLUB members have had their memberships extended until 1st February 1986. This means that the base of QLUB members is now much larger. That larger reader base means more people can contribute to make QLUB News the best QL newsletter or magazine you can find.

And to celebrate both the new, larger readership of QLUB and the first anniversary of the machine we're presenting our biggest issue yet — 24 pages packed with information, news, programs and product previews. Included in that 24 pages is a special eight-page pull-out spotlighting

new software for the QL — including business packages and programming aids.

We're also highly encouraged by all the letters you've sent in to our Programmer's Forum, QL Quarks and QLetters pages. Response to the competitions have also been overwhelming. Unfortunately we are unable to answer the letters individually as we don't have sufficient room to print them all. However we would like to take this opportunity to thank everyone who's written into QLUB News and invite you to continue doing so.

Our first competition alone generated more than 3,000 entries and hundreds of letters have poured into the QLUB News office. The competition results are published on Page 20, and the letters are

published on this page and the one opposite — as well as Pages 17 and 24.

So thank you all again, and happy anniversary.

WELCOME TO THE QLUB

THIS is the first of six newsletters that you will receive over the next twelve months.

Psion Software Support Limited also supply a comprehensive support service on QL Abacus, Archive, Easel & Quill, Qdos, SuperBASIC and any related peripherals — eg. Printers or memory expansion boards. Psion may be contacted by telephone on 01-723 0553 or by writing to:

Psion Software Support
22 Dorset Square
LONDON NW1

Q LETTERS

QL Super BASIC in Education

I WOULD first like to say how much I appreciate the improved format of the QLUB News. I would ask you not to forget the various educational functions of the QL. I am not only using the QL as a word processor, but I am finding it a very useful machine on which to write educational games. SuperBASIC gives great opportunities for writing good educational programmes.

You may be interested to know that I am writing this letter with the QL and a Dyneer Daisy DW20. I think that this makes a good combination for the small business.

David G. Burnett,
Missionary Orientation Centre,
Gerrards Cross.

Machine code book bind

I HAVE been trying to find a book with the Hex machine code for the 68008 without success.

Do you think that a chart or list could be included in the next QLUB News, also possibly an index for the QL Guide (not for Psion Packages).

D. T. Bardell,
Maidstone.

You'll find information on 68008 machine code books, as well as other QL books in next issue's QLibrary section. You'll also see that we have an on-going chart or list of books written on the QL to help you keep track of what's available. As to an index for the User Guide, the guide was designed to be added to and updated — making it difficult for any permanent index to remain current.

The Quest for QL disk drives

I AM pleased with the layout and information in Issue 2 and look forward to the next issue's advice on Sinclair Research peripherals.

No doubt you are aware of the very comprehensive range brought out by Quest International Computer Equipment Ltd.

From previous reading in QL advertising material I would have hoped for Sinclair equipment appearing through QLUB affording cheaper prices and discounts. The Quest equipment for Sinclair seems to be fully priced when compared with hardware already on the market for Apple and Commodore users.

On Page 2, top article, you invite requests. The most pressing needs to my way of thinking are the promised accounting packages (not just the Cash Trader one referred to on Page 3, but a full package that includes the same facilities as Tally I and II in the Quest catalogue, but at less cost than £174.50).

I cannot decide whether to go for floppy discs or the Winchester, managing

meanwhile with the Microdrive cartridge. Please advise.

An Expansion Console such as is shown in the Quest catalogue is also a fairly early "must," but £139 seems steep.

Finally, I would like clarification concerning the use of the F2 switch for the TV. If the Monitor switch F1 is used, my TV's show the two windows and seem to work on listing with a simple three or four lined program. If TV's can work with admittedly not so clear a picture, why the need for two switches?

Norman Smith,
Reepham, Norfolk.

AS you suggested, we are aware of the Quest system. As to discounts, QLUB works to get cheaper prices on all worthy QL equipment for its members. We haven't yet, however, concluded such an agreement with Quest.

You might consider, however, the CST disk system as a viable alternative (featured in this issue's Hardware Update). It allows you to use any standard disk drives available for the BBC Micro and can be used with the QL's own Qdos operating system, rather than CP/M 68K. The CST disk interface costs £149 and you can either buy disk drives from CST or get your own.

As to your request for cheaper accounting software, I invite you to look at this issue's special software pull-out. It includes details of the Sagesoft accounting package — which sells for under £100 and includes additional features over disk versions of the same software costing £300-£500.

And finally, on the question of monitors, the idea behind providing both switches is to

give you the optimum screen mode for the display you're using. While you can use the QL in 'monitor mode' on a TV, it isn't as easy to read as in 'TV mode.'

On using a bigger QL Quill

I READ with interest a letter in the QL User's magazine saying that QL Quill was not usable with documents of longer than 1,000 words. I have used it with a document of 3,500 words but found that at least 40 free sectors per 1,000 words is required on the Microdrive cartridge in the U.K., and feel that this information may be useful to other QL users.

J. M. Macdonald,
Rossendale, Lancs.

The Sinclair QL calendar

ISN'T it about time that you managed to synchronise the Sinclair calendar with that used by us mortals?

Issue 2 QLUB News was dated October, but arrived in late November. Most magazines are released some time during the month previous to their publishing date. Why the exception for Sinclair?

Getting an October magazine in November we were promised "We'll let you know more in November." Which November?

You also promise details on new Software for Christmas — QL Christmas wish list.

Which Christmas are we talking about, bearing in mind the Sinclair promises of delivery dates and their relation to reality?

When are we to get these updated versions of the Utilities so that I no longer have two or three hang-ups and restarts per A4 page on Quill?

I am Chairman of Queens Crescent Computer Club, and am beginning to get tired of the jibes I receive from Beeb and Amstrad owners.

William Jackson,
London NW1.

CONTRARY to what you might think, we work on the same calendar as you. There were some problems caused by the redesign of this journal which delayed the release of QLUB News Issue 2. As anyone who's ever worked in the printing industry will attest, things always take a good deal longer than you expect. And trying to arrive at the improved design of the Newsletter (which I hope you'll agree was worth the wait) caused

the inside copy to date somewhat from when it was originally written.

We offer our regrets to you and all QLUB members for the delays and have revamped the production schedules to ensure that these problems don't occur again.

A dose of QLUB concern

I AM writing to express my concerns at the way in which QLUB News appears to be developing and hope you will find a moment in which to consider them.

First, the continued optimism over timing.

The first issue was received on 3rd August 1984 and predicted completion of the ROM refit by the end of August 1984. My voucher was received on 12th October 1984.

The second issue, dated October 1984, was received on 16th November 1984 but still contains the statement "We'll let you know more in November." Will you?

Second, news continuity.

What happened to the software developers guide referred to in the first issue? If it is not going to be generally available soon, extracts from it might make future machine code articles more generally understandable.

Third, attitude.

Do phrases such as "Details can't be released yet — but that would spoil all the fun anyway" imply that someone on the editorial staff views the readership as immature school children rather than a group keen to learn more about their computer and how to make more effective use of it?

I hope that you will be able to develop the QLUB News into the valuable and reliable news sheet that I'm sure you intend it to be.

W.D. Brunt,
Crowthorne, Berks.

WHILE I can't comment on the delays you personally suffered with ROM refits, I can direct you to a similar letter about QLUB News delays (and my answer to it) above. As to news continuity and machine code articles, I suggest you turn to Page 15 where the Machine Code and More column provides guidance and programs in machine code each issue. The Software Developers Guide has been renamed the QL Technical Guide — further details can be found on Page 15.

And lastly, we DO NOT view readers as immature school children — but rather as mature individuals with a sense of humour who can appreciate the sometimes light-hearted tone adopted by this publication and take it in the spirit in which it's intended. Technical writing can become dry and boring if you don't adopt a lighter style in the writing. But we would never want to do so at

the expense of alienating readers with 'patronising' copy.

Miraculous modem mix-up

I AM interested in acquiring a modem for my QL and am slightly disconcerted by the apparent scarcity of a suitable device.

It was hinted at in the last QLUB newsletter that "Miracle Technology" were producing just exactly what I had in mind, but of course it isn't quite QL compatible at all-baud rates (like 300) and the auto answer extension won't work and it isn't quite ready yet anyway. Please could you let me know where I could get what I'm looking for? And where could I get a list of all the available wide area networks and bulletin boards?

Jim Timmis,
Twickenham.

THE Minor Miracle modem WILL work with the QL if you run it off the RS-232 interface provided by OEL's QCON communications interface. OEL also offers its own Prestel modem which we would highly recommend for use with the communications interface. More details on both devices were in Issue 3. As to bulletin board phone numbers, look in magazines such as Personal Computer World and Telelink, which publish phone numbers and operating hours for boards in every issue.

This is your page

White space is not a pretty sight, particularly in the eyes of an editor. Only you can prevent this dreaded disease from striking the letters' column of Sinclair QLUB News.

If we say or do something you disagree with, are surprised by or just want to sound off about, put pen to paper (or hands to keyboard) and let us know. Like it says at the top of this column, this is your page and it's up to you to make the most of it.

It isn't the only place in this magazine for your comments — there's the Programmer's Forum for your programming tips and queries, the QL Case Study to air your applications hints and the QL Quarks column to hear QL quips — but the letters page is the first place most people look for letters.

Write to:
The Editor, QLUB News
Sinclair Research
25 Willis Road
Cambridge CB1 2AQ

HARDWARE UPDATE

CST spins a winner

THIS issue's hardware update sees a change in format for the page. Until now, we've tried to feature a large number of new hardware products in order to quickly give you news of the widest possible choice of new QL-compatible equipment.

THE flood of new hardware developments for the QL has now reached the point where it would be impossible to list them all — and wouldn't provide you with much more information than you get in ads for the products. So Hardware Update is now shifting its focus in recognition of the need to not only tell you about new equipment and devices, but also to take a more in-depth look at some of the better ones. In future issues, Hardware Update will offer a detailed product report on devices Sinclair considers worthy of the QL. We start this new feature off with a spotlight on CST's Computamate disk interface and disk drive system for the QL.



Product: CST Computamate Q-Disk interface.

Price: £149 including VAT.

Description: General purpose disk interface and operating system extension for the QL.

Distributor: CST, Scotia Road, Burslem, Stoke-on-Trent, ST6 4DX (Phone: 0782 811711).

The CST interface unlocks the door to a wide variety of supplementary storage devices for the QL — from low-cost 200K 3.5 inch floppy disk drives right up to 5.25 inch dual floppy drives offering 1.5 Megabytes of storage in a single unit. The interface is designed to work with all industry-standard disk drives, and thus allows you maximum flexibility in choosing a storage medium to supplement the Microdrives built into the QL.

The interface is a long, rectangular circuit board which easily plugs into the left-hand side of the QL. When fitted, it sticks out about 2 inches from the end of the machine, but uses the same matt black plastic casing and thus fits in with the overall look and design of the machine.

The interface provides a multi-pin

standard disk drive socket at the left-hand side of the machine. You can plug a variety of drives into the socket, including all drives (provided they include their own power supply) built for the BBC Micro. In our evaluation, we tried both the dual floppy switchable 40/80 track drive system sold by CST for use with the interface and a standard BBC disk unit. Both worked faultlessly and provided a massive 1.5 Megabytes of storage.

CST's own disk drive prices start at £299 (including VAT) for a single 200K drive. Included in that price is the interface itself (which, as we said before, costs £149 if purchased by itself) and a power supply for the disk drive. The model we tested — with interface, 1.5 Megabyte dual double-sided 80-track 5.25 inch floppies — would cost only £574.

Perhaps even nicer than the pricing is the way the disk system is integrated with the QL's operating system. You'll be used to typing 'DIR MDV1_' to get a directory of Microdrive 1, so you should have no trouble with the 'DIR FLP1_' command needed to get a directory of floppy disk number 1. CST also provides a number of utilities with the disk drive which allow

you to transfer existing QL software and data files to disk without having to go through any complicated program modification procedures.

We successfully transferred QL Quill Version 2 onto floppy disk with no troubles at all. Once loaded onto the disk, everything loaded and ran from the disk drives in the same way it used to run from the Microdrives.

With two floppy disks, we were able to keep floppy disk 2 (FLP2_) for data and floppy disk 1 (FLP1_) for programs — just as they originally were on the Microdrives. Version 2 of the Psion software recognises commands for both the floppy disks and the Microdrives — so you can, for example, load up a file from Microdrive, edit it and then save the edited version to disk.

In summary, the disk interface and/or disk drives are a marvellous addition to an already powerful machine. CST has produced a system which existing QL users will find easy to learn, easy to afford and reliable to use. Disks won't replace the Microdrives, but they certainly will be a great help for any high-volume or high-speed work you want to do on the QL.

QLibrary

THERE are now lots of books available on how to make the most of your QL's hardware and software. To help those of you who have either bought or are considering buying one, QLibrary features interesting QL books from a variety of publishers.

The information here by no means constitutes book reviews, or recommendations by Sinclair Research — but rather information we're passing on. Like a real library, we'll show you a large stock of books and it will be up to you to decide for yourself what you really want to read.

THIS issue's featured work comes from Longman's 'Working with the Sinclair QL' series, edited by Roy Atherton, who wrote the beginners section of the QL Users Guide and is the Head of Information Technology at the Bulmershe College of Higher Education in Reading. The series goes under a silver and black banner in the top left-hand corner of each book designating it as part of the 'Working with the Sinclair QL' series.

The first book in this series to make its way to our desk was Roy Atherton's *Good Programming with QL SuperBASIC*. The book is in a 'beginners start here' style which opens with details on how to format Microdrives, what each of the keys on the keyboard is supposed to do and how the QL's various screens are laid out.

The structure of the book is very logical, and assumes little forehand knowledge of programming concepts or methodology. Program listings (which are printed from a daisywheel printer and include an indication that the 'ENTER' key has to be pressed after each line in the earlier part of the book) are clear and often accompanied by structure diagrams or flow charts.

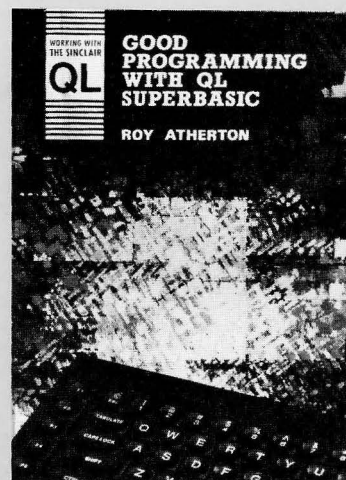
Chapter Two begins by introducing you to essential programming principles and acquaints you with commands such as PRINT, RND, LET and REMARK. The program examples are mostly of the order of 'how to simulate the roll of a dice' or

'how to get the computer to print your name on-screen' — pretty standard stuff but they give you the general idea.

You are then introduced to control structures in Chapter Three (things such as FOR...END FOR, FOR...NEXT and REPEAT) and given several program examples in both listing and flow diagram form. These program examples won't cause any major geological disturbances or do anything spectacular, but there's no need to blind beginners with colour and graphics before they really know what it's all about.

Then, after you've spent a chapter looking at data structures and another on a 'simple case study' you actually get down to using your newly-acquired skills to produce some graphics. Chapter Six's graphics programs give you tips on using colour, drawing ellipses and making the best of turtle graphics commands.

Without going into too many details on the rest of the chapters, you get tips on mathematical operators, file handling, windows and advanced control structures. The book is a bit dry, but it's thorough and will give anyone a well-rounded introduction to programming in SuperBASIC on the QL. The Longman QL books — including *Good Programming with QL SuperBASIC* — can be ordered from the Longman Group Limited, Longman House, Burnt Mill, Harlow, Essex CM20 2JE, England.



OFF THE SHELF

IN response to many requests from readers, we have compiled a list of books and book series about the QL. We will continue to add more books as they are published. If you know of any worthwhile QL books not included on this list, or which we have not featured, let us know and we'll try to include them in future issues. The books or series marked with an asterisk (*) have been featured in QLUB News.

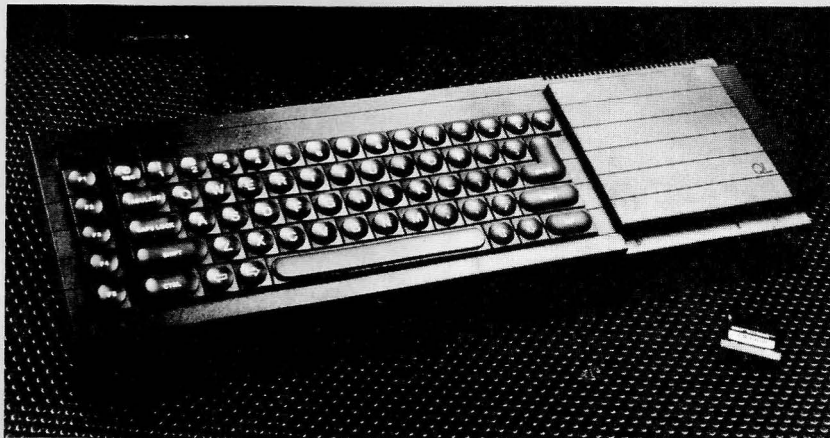
The Sinclair QL Series — five books in all; 'Introducing the Sinclair QL', 'Introduction to SuperBASIC on the QL', 'Advanced Programming on the QL', 'Desktop Computing with the Sinclair QL', 'Word Processing on the Sinclair QL' — Available from TBS, 38 Hockerill Street, Bishop's Stortford, Hertfordshire for £6.95 each. *(Featured in Issue 2.)

The Blueprint QL series — five books, one covering each of the four software applications which come with the machine and one discussing how to exchange information between them. Available from Century Communications Ltd., Portland House, 12-13 Creek Street, London W1V 5LE and selling for between £6.95 and £7.95. *(Featured in Issue 3).

'The Working Sinclair QL' — A programming guide written by David Lawrence and sold as part of a QL series by Sunshine books. Available through Sunshine book distributors for £6.95.

'Quantum Theory: A Guide to the Sinclair QL' — This Century book is written by three people (Jeremy San, Fouad Katan and Simon Rockman) and claims to be one of the first books written using a 'production' QL. It's available from Century Books at £5.95.

'A QL Compendium' — A good old games listings book by Martin Gandoff and Robin King. It's published by Addison Wesley and sells for £7.95.



INTERFACE INQUEST

Joystick Jottings

WITH the current craze for joystick-controlled software, it's good to know that the QL can join in the fun. Two of the sockets at the back of the machine are the key to all this, as they allow joysticks to be plugged into the QL and be read by all sorts of software — even Basic programs!

The sockets are the ones marked 'CTL1' and 'CTL2', and they share the same British Telecom-style sockets. They are wired up in such a way that they appear to be 'in parallel' with certain keys on the keyboard. CTL1 conforms to the four 'arrow' cursor keys and the space bar, and CTL2 corresponds to the five function keys. This means that if we wrote a piece of software which used one or both of these groups of keys to control it, then exactly the same effect could be obtained by plugging a joystick into the corresponding socket and using that instead.

The wiring diagram of each socket has so far not been published. But we have rectified that with the diagrams shown in Fig. 1. Using these designations, you can

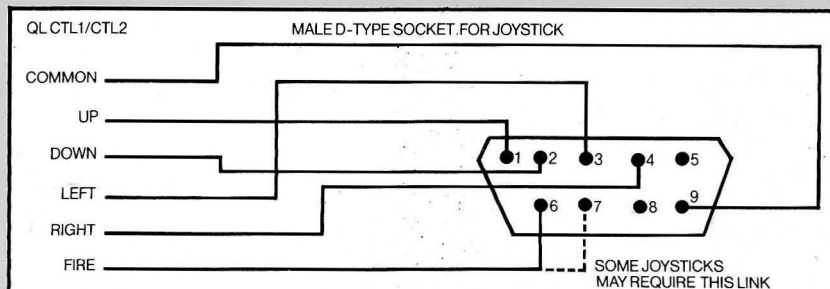


Fig. 1.

| CTL 1 | | | CTL 2 | | |
|-------|--------|---------------------------|-------|--------|--------------------------|
| pin | name | function | pin | name | function |
| 1 | Common | signal ground | 1 | Common | signal ground |
| 2 | Left | acts like left arrow | 2 | F1 | acts like function key 1 |
| 3 | Right | acts like right arrow | 3 | F3 | acts like function key 3 |
| 4 | Down | acts like down key | 4 | F2 | acts like function key 2 |
| 5 | Up | acts like up key | 5 | F4 | acts like function key 4 |
| 6 | Space | acts like the fire button | 6 | F5 | acts like function key 5 |

Fig. 2.

make up your own joystick lead and get programming. As each joystick movement simulates a keypress, we must ensure that any device we plug into these sockets is of the 'switching' type rather than the 'potentiometer' type used by some micro manufacturers. Luckily, most of the joysticks around are of this type, so it shouldn't be too difficult to find a

suitable one.

Figures 1 and 2 show the appropriate connections, which apply to both sockets, to allow a normal 9-pin 'D' socketed joystick to be used in place of the keyboard.

Now, because of the way these ports are configured inside the QL, hitting either one of the keys or moving the joystick in the designated direction will cause the desired effect to be created. So, if we're using INKEY\$ in a Basic program, waiting for 'space' to be hit:

```
REPEAT waitpace:IF INKEY$="" :EXIT
waitpace
```

then pressing the spacebar or 'fire' on the joystick in CTL1 will exit the loop. This is advantageous, as it means that a program written for the keyboard will run, without alteration, with a joystick.

Likewise, the KEYROW function can be used to see if a key is being pressed or if the joystick is in a particular position:

```
REPEAT waitfire:IF KEYROW(0)&&32:
EXIT waitfire
```

This loop will terminate when either F5 or the 'fire' button on joystick two is pressed. The listing (at left) is an example of how to use these commands with the joysticks, and moves a character around the screen under the user's control. Notice how it uses the INKEY\$ function with a timeout of -1 to read the keyboard or joystick, followed by SELECT to determine the action to be taken. The INKEY\$ timeout specifies the number of 20ms intervals that the function will wait before returning an answer. By using -1 instead of a positive value here, we're saying 'wait for ever'.

```
100 MODE 0
110 WINDOW 440,180,36,20:BORDER 1,7:PAPER 0
120 OPEN#3,scr_440x12a36x200
130 PAPER#3,2:INK#3,7:BORDER#3,1,7
140 PRINT#3:"Use cursor keys or joystick 1 to move -
space / fire to change character";
150 PAN#3,2
160 char$="0"
170 x=38:y=10
180 REPEAT main
190 AT y,x:PRINT char$;
200 keyin=CODE(INKEY$(-1))
210 SELECT ON keyin
220 =192:left:REMark left cursor key / joystick 1
230 =200:right:REMark right cursor key / joystick 1
240 =208:up:REMark up cursor key / joystick 1
250 =216:down:REMark down cursor key / joystick 1
260 =32:change:REMark space bar / fire button
270 END SELECT
280 END REPEAT main
290 DEFINE PROCEDURE left
300 clchr
310 x=x-1:IF x=-1:x=71
320 END DEFINE
330 DEFINE PROCEDURE right
340 clchr
350 x=x+1:IF x=72:x=0
360 END DEFINE
370 DEFINE PROCEDURE up
380 clchr
390 y=y-1:IF y=-1:y=16
400 END DEFINE
410 DEFINE PROCEDURE down
420 clchr
430 y=y+1:IF y=17:y=0
440 END DEFINE
450 DEFINE PROCEDURE clchr
460 AT y,x:PRINT " ";
470 END DEFINE
480 DEFINE PROCEDURE change
490 LOCAL a
500 a=CODE(char$):a=a+1:IF a=58:a=48
510 char$=CHR$(a)
520 END DEFINE
```


COMMUNICATIONS CORNER

Are you a fan of QLAN?

THE QLAN local area network built into the QL is a fast means of communication between two or more QLs connected via very simple 'two-wire' network leads. The network is treated by Qdos (and therefore by SuperBASIC) as another input and output device, identical as far as it's concerned to Microdrives, printers and so on.

On the face of it then, the QLAN seems to be very similar to the RS-232 ports, but there is one major difference. As the network only use two wires, the 'hand-shaking' protocols which ensure that data is not lost must be dealt with in software, rather than the RS-232's hardwired system. This results in the network dealing with data in 256 byte blocks rather than individual characters. If you're transferring large amounts of data, such as complete

files, then this can be an advantage, but it's not recommended if you want to send small amounts of data.

The only way of ensuring that a quantity of information less than 256 bytes long is transferred across the net is closing the channel to the network as soon as the data is transmitted. This is done implicitly by operating system commands such as COPY, SAVE and so on, but not if you've explicitly opened the channel using OPEN#. Unfortunately, OPEN# is really the only means you have to transfer data from programs, so you'd need little routines like

```
OPEN#3,neto_2
PRINT#3;<our_data>;
CLOSE#3
```

each time you wanted to send some information. This is not ideal and means that you can't have a viable two-way communication system using the net, as you very rarely speak in sentences of exactly 256 characters!

If you wanted to implement this sort of system, you would need to use the serial ports. This doesn't mean that the network can't be used, as you can use it as a general purpose filing system which allows you to use a common set of peripherals between a group of QLs. You can, for example, have a printer, CST disc drives and so on on one QL and let other QLs use these peripherals if you write control program to inform the 'master' QL of what you want to do.

The two programs here together form a very basic control program for the network, allowing up to 63 QLs to use the Microdrives and printer connected to a further QL. This 'master' QL is given station number 1, and runs the program in listing 1 permanently. This examines the network, waiting for certain characters, and then it acts upon those characters. The NPRINT procedure, in listing 2, should be resident in each machine which wants to use the master QL. Whenever a station wants to print a file, type

NPRINT 'P',<filename>

and that file will be sent to the master QL which will print it. If a station wants to read a file from the master's Microdrives, then type

NPRINT 'R',<filename>

and you will be asked for the device to read it at the station end. To save a file onto the master's drives, type

NPRINT 'S',<filename>

and the specified file will be saved. Notice that you use

PEEK(163895)

on line 120 of the second program. This address is the SV_NETNR system variable, and it contains the station number of that QL.

There's a great deal of scope for improvement in these two programs, but they should give you the basic idea of how to use the network.

QLAN Control program - Listing 1

```
100 MODE 0
110 PRINT 'QLAN Control program'
120 pr$=seric:REMark this is the printer device
130 NET 1
140 REPEAT loop
150 OPEN#3,neti_1:REMark this is 'general listen' mode
160 op$=INKEY$(#3,-1):thatnet=CODE(INKEY$(#3,-1))
170 IF op$='R' OR op$='S':INPUT#3;f$
180 CLOSE#3
190 PRINT 'Operation 'op$' from station 'thatnet)
200 op=CODE(op$)
210 SELECT ON op
220 =CODE('P')
230 COPY_N 'neti_'&thatnet,pr$
240 =CODE('R')
250 COPY_N f$,'neto_'&thatnet
260 =CODE('S')
270 COPY_N 'neti_'&thatnet,f$
280 END SELECT
290 END REPEAT loop
```

QLAN peripheral control procedure - Listing 2

```
100 DEFINE PROCEDURE NPRINT(op,file)
110 LOCAL thisnet,a,f$
120 thisnet=PEEK(163895)
130 IF op='P' OR op='S' OR op='R'
140 OPEN#3,neto_1
150 PRINT#3;op;CHR$(thisnet);
160 ELSE
170 PRINT 'Illegal Operation!'
180 RETURN
190 END IF
200 a=CODE(op):SELECT ON a
210 =CODE('R')
220 INPUT#0;'Read into: 'f$
230 PRINT#3;file
240 CLOSE#3
250 COPY_N neti_1,f$
260 =CODE('S')
270 PRINT#3;file
280 CLOSE#3
290 COPY_N file,neto_1
300 =CODE('P')
310 CLOSE#3
320 COPY_N file,neto_1
330 END SELECT
340 END DEFINE
```

Quantum Leaps: The QL Case Study

THE QL is used by many different people for many different things. They either tailor existing software or write their own programs to accomplish specific tasks with the machine. Each of these adaptations is, in its own way, a quantum leap. This column is devoted to highlighting such uses of the QL.

OVER the next two years Sinclair Research is giving several hundred QLs to students at Strathclyde University in an extended experiment.

Meanwhile, Strathclyde's importance as a centre for research on artificial intelligence (AI), as well as its close links with the nearby Turing Institute (itself opened late last year as a government-supported centre for work on AI), seems destined to involve the QL in some of the most high-level computer research in the world.

"At the moment we've got 100 QLs" says Andrew McGettrick, head of Strathclyde's Computer Science Department. "They're being given to MSc students on an Information Technology course."

"The students will use the Psion spreadsheet and database programs provided with the QL, but an important part of their responsibility in this first year of the scheme is to produce a report for Sinclair."

A network, extending over large parts of the campus, will give individual students the chance to access a store of lecture notes, assessments and study material — all contained on the university's main computer. At the same time, the university will be setting up a central database of course-related information to which each QL can be linked.

For access from the halls of residence, students will need some form of modem, but most of the work will be local access, perhaps from a laboratory, and no kind of interface will be necessary — students will be able to bring their QLs, plug them into the network and upload and download the material via the RS232 port.

Andrew is hoping that all this will be ready for next summer, when Sinclair Research will be giving the university another 425 QLs — to be handed out to students of computer science. These students will be doing all their studying — including Pascal programming — on the QL.

"We hope that by the end of the decade every undergraduate will have a QL that he can take home, hook up to the network, and use as his own," says the university's Mr. Richard Kingsley.

"The QL is a very powerful computer in a very small box, so students can carry around tremendous computing power in their briefcases. That obviously gives us all sorts of ideas for teaching packages — for instance, we want students to be able to follow programs dynamically on screen, using the QL's windowing facilities."

We've nearly finished a self-teach Basic package which runs programs for you and monitors what they do while they're running — checking for mistakes and telling you when you've done something wrong. And we hope to do similar things for assembly code and for Pascal.

"The superb graphics facilities on the QL should be used to the full, so we're hoping that our plan for networked lecture notes can include self-running programs which generate illustrations for the text."

The university is also likely to use the QL's Lisp implementation for research on "expert systems" — databases of information which the computer presents in a helpful and interactive way, comparable to that of a sympathetic human expert — and is contemplating the use of CST's IEEE interface in robotics work.

QLs are also to be found at the nearby research centre for work on Artificial Intelligence, the Turing Institute, which is affiliated to the university.

The Institute's Assistant Director of Advanced Studies, Judith Richards, explained that the QL was being used to run an expert system demonstrator. "But, of course, the QL will be capable of handling much more than just a small demo shell, and because it's inexpensive will form a good basis for people who want experience in expert systems without making a large investment."

"Provided you don't need a really huge database, the applications possible on the QL will be fairly varied."

Mr. Kingsley explained how the Turing Institute's demonstrator works on the QL. "In lots of systems, when you type HELP, you're given only off-the-shelf aid. The Turing demonstrator, though, gauges how well you're progressing. If you prove to be quite adept, you'll be given only superficial help; but the system will sense if you're

floundering, and provide more gentle prompts, and eventually quite detailed technical help. It learns about you as you go along."

And would commercially-available software develop out of the Strathclyde venture?

"Oh, undoubtedly," says Mr. Kingsley. "Nowadays, lots of universities are involved in marketing software, and our aim would be to sell the material — at the very least to educational establishments, and preferably to other people, too."

Judith Richards underlines that commitment to the commercial sector. "The QL has the capability of producing a great number of business and expert system packages, and if the university works on them it's likely that there will be at least some competitive packages around for other people to take into commercial exploitation."

The first fruits of that ambition is likely to be the basic course, says Mr. Kingsley. "After that possibly the teaching package for assembly work, and perhaps a program for teaching logic diagrams."

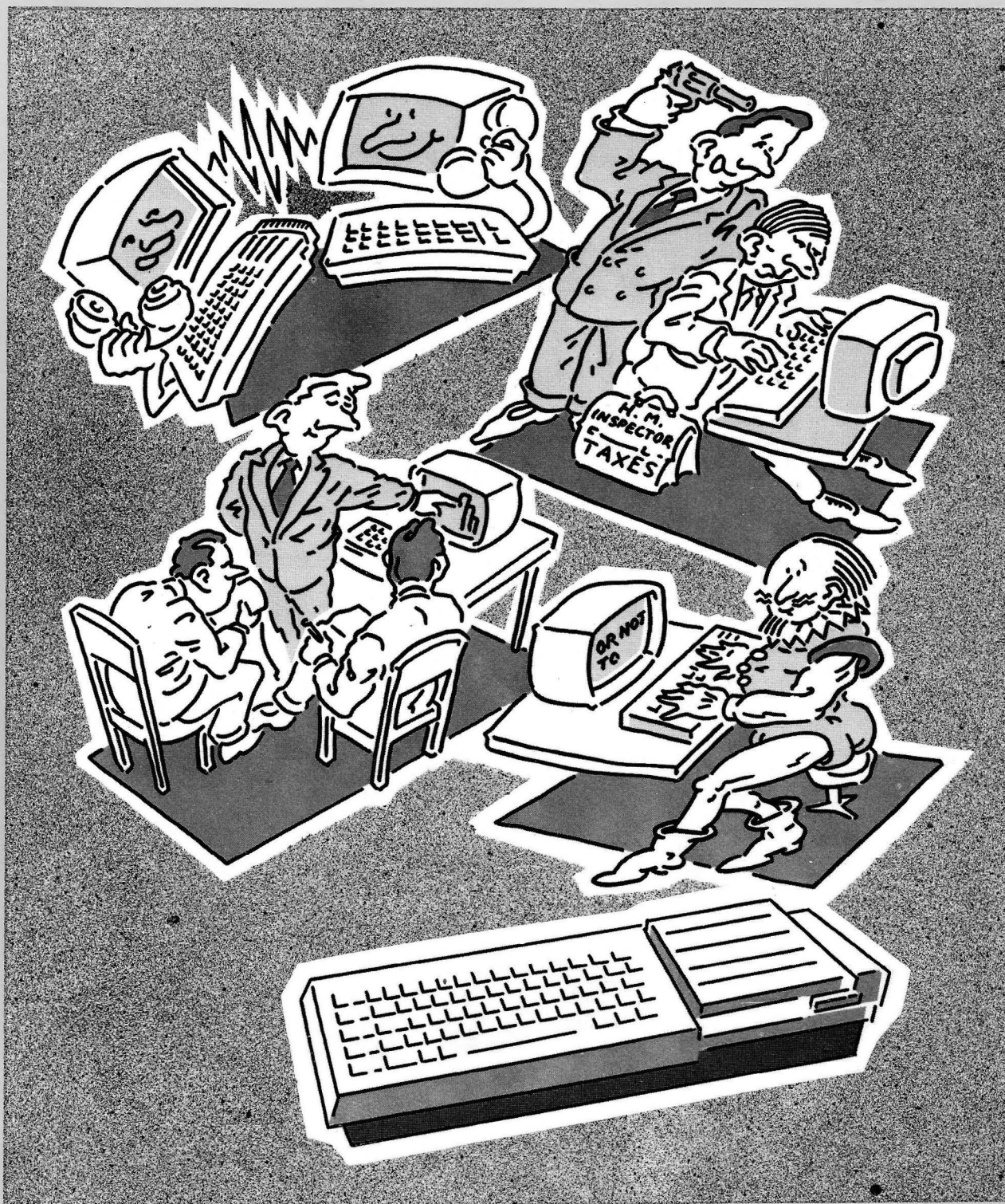
Not surprising, then, that our own Nigel Searle says that this degree of forward planning and software spin-off puts the Strathclyde plan "head and shoulders" above similar proposals from universities throughout Britain.

But wouldn't this use of computers as a teaching aid turn the university into a very impersonal place?

Not so, according to Professor James Alty, head of Strathclyde's Computer Science Department and a Director of the Turing Institute. "Almost the opposite could be argued. The introduction of machines will liberate staff and students from purely routine work, so that the number of tutorials and seminars could actually be increased."

If you have any unique application idea for your QL which you'd like to share on these pages, write to:

**Quantum Leaps:
The QL Case Study
The Editor QLUB NEWS
Sinclair Research, 25 Willis
Road, Cambridge CB1 2AQ**



- Cashing in with QL Cash Trader
- Accounting on your QL with Sage
- Version 2 Psion software
- Software developer's news
- Project planning and decision making
- QL Toolkit

SOFTWARE SELECTIONS

WELCOME to the first QLUB news special pull-out. In it, we'll focus on the best of the flood of new software recently released for the QL in a more in-depth manner than was previously possible in our one-page software update. This special pull-out includes features on accounting software, project planning programs, development tools and more.

The articles contained in this section don't constitute reviews, but merely factual descriptions of the products. However, you can be assured that the products wouldn't be featured unless we felt they were worthy of your attention. At the top of each article, you'll find a box giving you ordering details, prices and discounts to QLUB members, the name of the company

responsible for the software and the equipment you'll need to run it (in most cases this will simply be a standard QL, however in instances where a printer or dedicated monitor may significantly enhance your use of the program, we'll point this out).

You'll note that we're not only featuring new software in this section, but also news of updates to the existing Psion software — which should by now have been sent to all QLUB members who signed up to get them last year. Also included in the section on the Version 2 Psion software are a few notes on converting old QL Quill, Abacus, Archive and Easel files to run under the Version 2 software.

Perhaps the best software news of all is

that the price of Microdrive cartridges has come down, which will mean that not only can more of you write and develop your own specialised software on Microdrive, but also that software houses which had been holding off release of new programs for the QL until the Microdrive cartridge price dropped can now enter the market with new vigour.

If you like the pull-out idea and enjoy this one, write and let us know. If you want more supplements like this one, then we'll print them — after all, it's your QLUB. To make sure your comments get to the Supplements Editor, write to:

**QLUB Supplements
The Editor
QLUB News
Sinclair Research Ltd.
25 Willis Road
Cambridge CB1 2AQ**

QL-Cash Trader opens for business

Name: QL Cash Trader
Price: £69.95 including VAT, £54.95 to QLUB members.

Author: Accounting Software Ltd
Publisher: Sinclair Research Ltd
Ordering Address: Write to Sinclair Research, Stanhope Road, Camberley or telephone 0276 685311.

QL CASH Trader is an easy-to-use cash-based accounting program from Accounting Software (a subsidiary of Quest International). QL Cash Trader requires little accountancy skill to operate and keep a record of all transactions, instantly updates the profit/loss statement and uses the QL's graphics capabilities to present information in a clear and easily understood form.

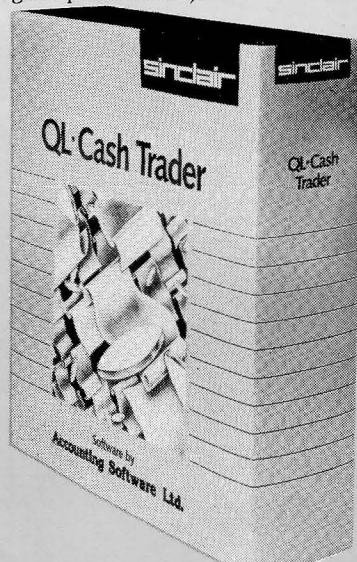
Using QL Cash Trader you can use a single screen to enter a wide range of unrelated transactions from credit sales and purchase to business expenses and cash movements. Using the QL's unique 'windowing' facility, QL Cash Trader is able continuously to display updated balance sheet as well as profit and loss figures while new transactions are being entered into the system.

The program comes on three Microdrive cartridges — QL Cash Trader 'Boot' and QL Cash Trader 'Program'. A large chunk of QL Cash Trader is loaded into the QL's memory using 'Boot' and then the program prompts you to insert 'Program', which loads in the rest from Microdrive as it's needed. An on-screen prompt also warns you when data or program information is being loaded, so

you can feel confident that newcomers to the QL will be able to use the program without a great deal of special training.

QL Cash Trader provides a tutorial and a series of examples with the package which step you through the basic operations of the software, and ensures that operators understand the program before they start using it.

A 200-page manual, written in simple and non-technical language, includes sample set-up instructions for eight different businesses. As we mentioned at the beginning of this feature, QL Cash Trader is available to QLUB members at a specially-discounted price of £54.95 (regular price £69.95).



CONTENTS

High Finance for small business . . . 10
Trying to get the books straight, and hoping the QL will lend a hand? Look to our report on QL Cash Trader.

Take a number, please. 11
SageSoft's accounting package for the QL sends the cost of professional business accounting software tumbling.

Something Old, Something New . 12-13
A word or two on the new version of QL Quill — as well as notes on how to convert files from Version 1.

QL Decision Maker+QL Project Planner 14
These two packages will assist both forward project planning and the rational decision-making process.

Developments and Dreams 15
News on new development software for the QL, along with details on where to get it.

Toolkit tales 16
An examination of this new programming development aid for SuperBASIC and machine code programmers.

QL-INTEGRATED ACCOUNTS

by Sagesoft

Name: QL Integrated Accounts
Price: £89.95 including VAT
Publisher: Sagesoft
Available from: Sinclair Research — due for imminent release

ONE of the most 'taxing' (if you'll excuse the pun) business applications any computer can take on is accounting work. It often requires complex calculations and the handling of large amounts of data — from sales invoices to bank receipts — and in the past has often meant that programmers skimped on ease-of-use in order to provide all the features necessary to provide a powerful program.

Sage's QL Integrated Accounts takes accounting software one step further. Not only does it provide all the facilities you get in Sage Software on the IBM PC and ACT Apricot versions, but the QL windowing and multi-tasking facilities encourage a much more 'friendly' and easier-to-use approach for only £89.95.

Let's take a quick 'tour' of the package's facilities. Because of the totally integrated nature of the software, entries made to sales and purchase ledgers automatically up-date the general or nominal ledger and all other appropriate ledgers. This means your books are always up-to-date and in balance. It also means you can quickly and easily produce a trial balance and management accounts, profit and loss statement and current balance sheet.

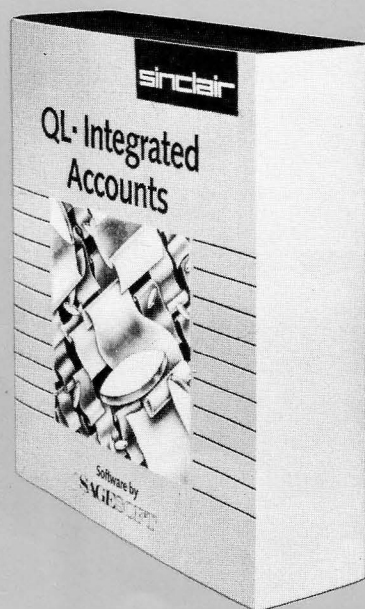
The integration involved in QL accounts follows a growing trend among software developers to 'free-up' data so that it can be used to as much of a program is possible. This even extends to the new versions of the Psion Software, which allows you to more easily 'import' and 'export' data between the various packages (QL Quill, Abacus, Archive and Easel) so that, for example, information from an Abacus Spreadsheet can be used to provide data for an Easel graph.

Aside from the lack of friendliness, and the lack of integration in many other accounting packages, there is also usually a problem with the documentation or manuals — which tend to be both tortuous

and complex. This is often the result of having 'unfriendly' programs to begin with — although the scientific tone adopted by many manuals make the subject more dreary than it need be.

QL Integrated Accounts looks to get round the problems of simple and clear documentation for a complex product by providing a tutorial which runs through most of the basic accountancy disciplines involved in using integrated accounts — and then allows the user to practice them.

So with the clear instructions and tremendous power in this integrated package, you should be able to put your accounts in order without exceeding your budget.



QL Integrated Accounts comes with a complete tutorial package for easy learning.



HM Inspector of Taxes.

VERSION 2 SOFTWARE

VERSION 2 of the Psion applications software for the QL will now have been sent to all QLUB members who signed up for it. This comprises updated versions of QL Quill, Archive, Abacus and Easel on Microdrive.

For those of you who have only just started receiving QLUB News, or who haven't delved deeply into your new versions of the software yet, it might be helpful to go through each of the packages

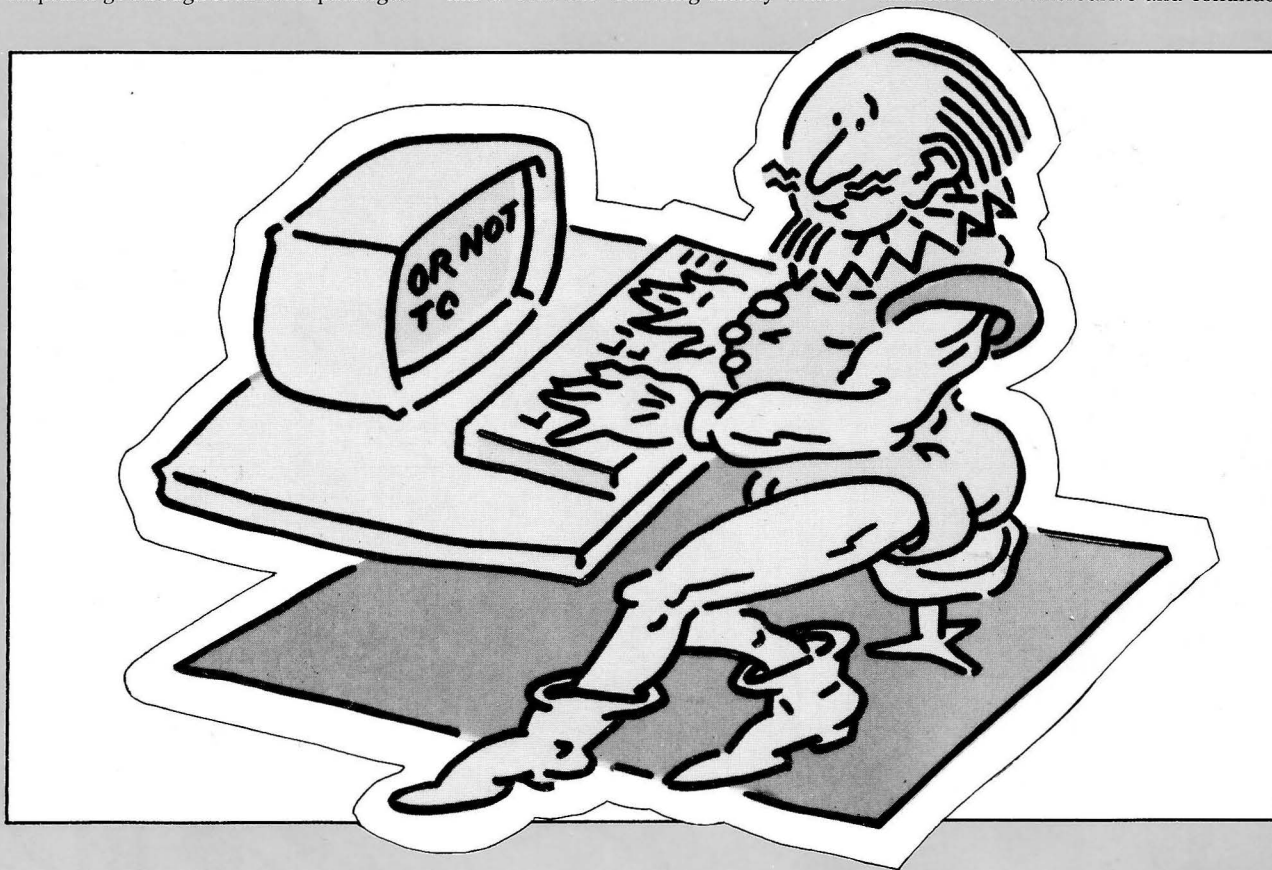
and discuss the additional features or changes in design.

We'll be taking that in-depth look over the next couple of issues. To begin with, we take a detailed look at the new version of QL Quill.

QL Quill — the QL's word-processor — has probably benefited the most from the software updating. Not only does the program load a good deal faster, but it also has a 'ram-file' buffering facility which

means that for short documents (anything under about 1000 words) QL Quill doesn't even touch Microdrive 2 until you actually take the action of naming or saving the document.

What QL Quill 2 actually does is to set up a temporary file in the QL's RAM memory. When that file is filled up (at about the 1000 word mark) the program asks you whether you want to save the information to Microdrive and continue,



THE CONVERT UTILITY

VERSION 2.0 of the install_BAS program has been modified to offer a wide range of printer options. This means that it is not compatible with install_DAT files created with version 1.0. A conversion program, convert_BAS, is supplied on the QL Quill and QL Abacus cartridges to convert Version 1.0 install_DAT files so that they are readable by the version 2.0 installation program.

Put a cartridge containing a copy of convert_BAS in Microdrive 1 and a cartridge containing your Version 1.0 install_DAT file in Microdrive 2. Run the program by typing:
`LRUNMDV1_CONVERT_BAS`

The program pauses to allow you to change the cartridge in Microdrive 1 if you want. Press ENTER when you are ready to continue. It then reads the

install_DAT file in Microdrive 2 and writes the new version to Microdrive 1. Note that the new version will replace any install_DAT file on this cartridge. You can then, if necessary, copy the new install_DAT file to another cartridge.

WARE IS OUT NOW!

print the document out or just save it.

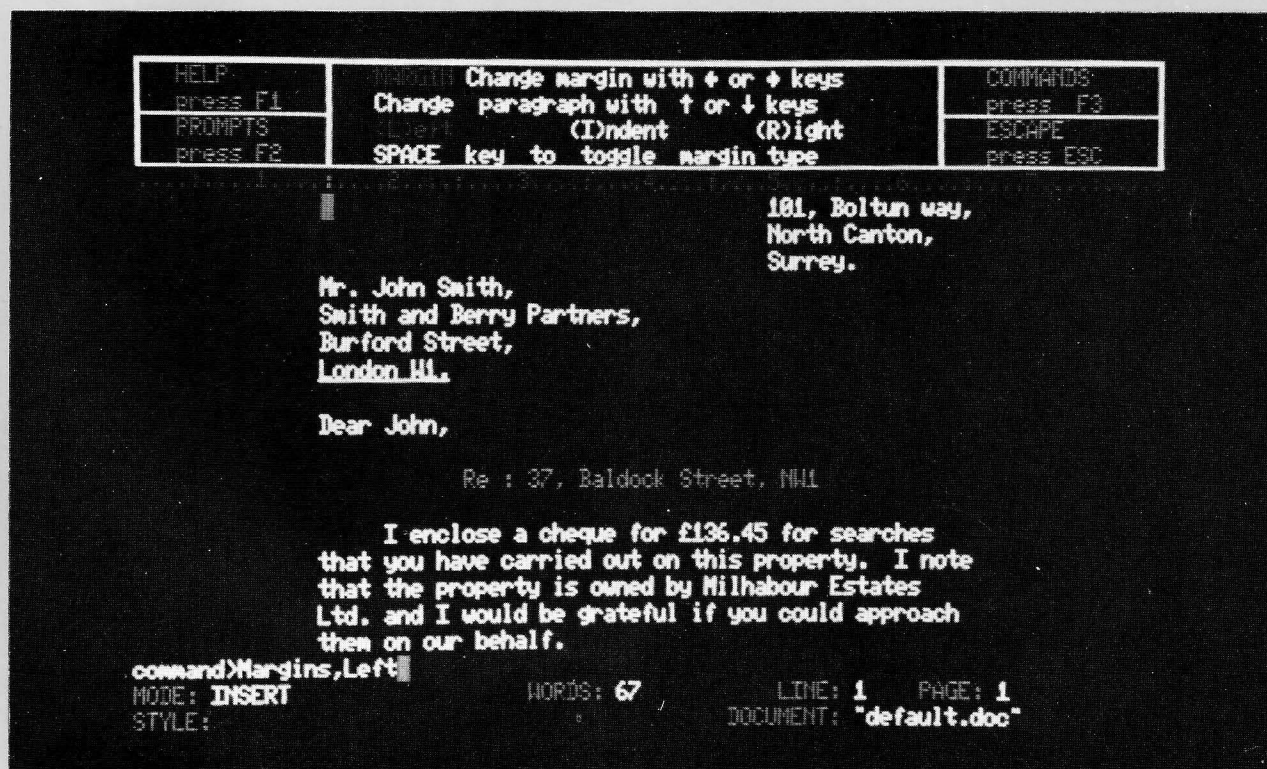
There's also a good deal less accessing of Microdrive 1 to get command and help file information in this version of the software. In the same way as QL Quill 2 sets up room for a 'RAM-file' it also sets aside some room for keeping the commonly-used commands in memory instead of having to pull them off Microdrive. This makes the new version of QL Quill much faster all-round — both in giving com-

mands and in entering text.

The device-independent nature of QL Quill has also been enhanced in the second edition of this software. Not only can you now easily load 'doc' files from Microdrive, but commands for disk drives (FLP1 or FLP2), the serial ports (SER1 or SER2) and exported files (EXP) are also fully implemented and files can either be loaded from or printed to any of those devices.

QL Quill can also now handle very large documents — we tried one of about 4500 words and found that it still could be saved as a QL Quill file (although this is not recommended as it makes the file somewhat unwieldy to work with).

All around, the new version of QL Quill is well worth having as it enhances and builds on the strengths of the Version 1 software and adds in some useful new features.



THE SCREEN CONVERT UTILITY

VERSION 2.0 of QL Archive allows you to choose the ink colour for the value of a variable shown in a screen layout. It sets the colour to the ink colour that is current at the time the space for the variable is reserved. In Version 1.0 the colour could not be selected. This means that screen layout files created with Version 1.0 cannot be loaded by Version 2.0.

A conversion program — SCRCON_BAS — is supplied on the QL Archive cart-

ridge to convert Version 1.0 screen layout files so they are readable by Version 2.0. Put a cartridge containing a copy of SCRCON_BAS in Microdrive 1 and a cartridge containing your Version 1.0 screen layout file(s) in Microdrive 2. Run the program by typing.

LRUNMDV1 SCRCON_BAS

Type in the name of a screen layout file (the program assumes an extension of _SCN) and press ENTER when you are ready to continue. It then pauses to allow

you to change the cartridge in Microdrive 1 if you want. Press ENTER when you are ready to continue. It then reads the old version of the file from Microdrive 2 and writes the new version, with the same name, to Microdrive 1. Note that the new version will replace any file with the same name on this cartridge. You can then, if necessary, run the program again to convert another file.

Planning and Projecting

Names: QL Decision Maker
QL Project Planner

RSP Price: £39.95 each

Available from: Sinclair Research — due for imminent release

Financial decisions and project planning are activities often difficult to conduct in the rational, calculated manner that they should be carried out.

'Figuring the odds' in business and attempting to produce reasonable schedules for projects is made much easier by the development of two new packages for the QL: QL Decision Maker and QL Project Planner. The first package allows you to quantify how important various options are in a decision-making process and arrive at a logical conclusion based on percentages and values you assign to each possible outcome, while the second lets you see what happens to various stages of work in meeting a deadline, when the time for one or other of the jobs in the project changes.

QL Decision Maker — introduces you to the concept of 'Decision Trees', which operate by taking a given decision as a starting point and then growing various 'branches' representing the various possible outcomes of that decision. You also have to give each possible outcome given a possible negative or positive numeric value (—8 being the least desirable outcome and +8 being the most desirable outcome) and then a decimal number (or a letter representing that decimal number) representing the percentage likelihood of that outcome happening.

The program will then develop a list of



potential 'payoffs' for each of the possible outcomes and assist you in arriving at the right decision.

QL Project Planner — helps you break down each activity in a given project into its constituent parts, so it can be analysed and 'rationalised'. These projects usually start by having a particular goal which must be accomplished and a deadline to accomplish it.

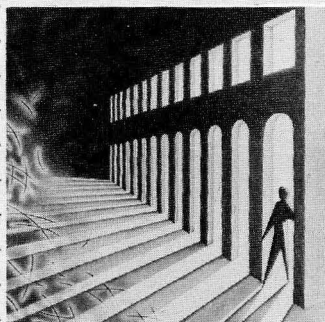
QL Project Planner helps you put all those thoughts together by first getting you to specify — in detail — each of the jobs involved in achieving a given task,

the duration of those tasks and the prerequisites to accomplishing them. When you've entered all that information, QL Project Planner will give you a 'wall chart' to show the life of the project and allow you to see in detail how long each activity will take and where there are overlaps (such as in the hiring of staff and buying of materials if you're putting a building together).

QL Project Planner is a valuable aid in developing project profiles and in its implementation on the QL is very easy to use.

sinclair

QL· Project Planner

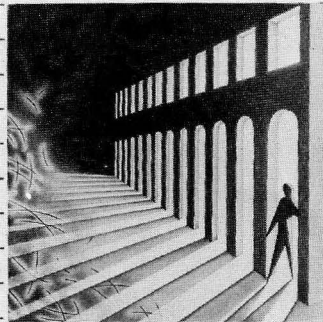


Software by

Triptych Publishing Ltd

sinclair

QL· Decision Maker



Software by

Triptych Publishing Ltd

Interesting developments

IN this special section, you will have seen that a large number of software houses and individuals are already writing or releasing products for use on the QL. In order to speed that process, not only have we cut the prices of Microdrive cartridges, but we have also developed many tools for programmers wanting to write for the QL.

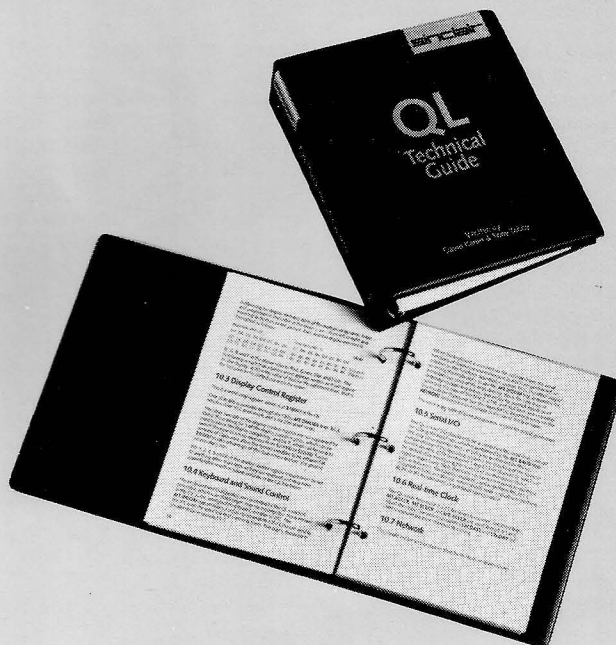
We are now making information and software tools for the QL much more generally available in an attempt to ensure maximum compatibility among new QL software and hardware peripherals.

The QL Technical Guide is a complete technical guide to the QL which includes the full specification of the Qdos operating system. Parts of this specification have been available for some time in the form of photocopied sheets. Now the complete specification has been professionally collated and expanded to provide the information needed by software developers. It will also be useful for anyone developing hardware peripherals for the QL. (The guide can be obtained by sending £14.95 (including VAT) to Sinclair Research, Stanhope Road, Camberley, Surrey GU15 3PS.)

The SuperBASIC production kit is available to writers of commercial programs in SuperBASIC. It takes the form of a set of machine code routines which extend the facilities of SuperBASIC and which, under licence from Sinclair, can be incorporated into commercial SuperBASIC programs.

It allows fast image-loading of basic programs, fixed decimal point conversion, error-trapped input/output, random file access and copy protection. The package is loaded from Microdrive cartridge and remains accessible to SuperBASIC programs until the machine is reset.

The SuperBASIC production kit is available direct from The Software Dept., Sinclair Research Ltd., 25 Willis Rd., Cambridge CB1 2AQ and costs £100. Written enquiries only please.



The QL Technical Guide.

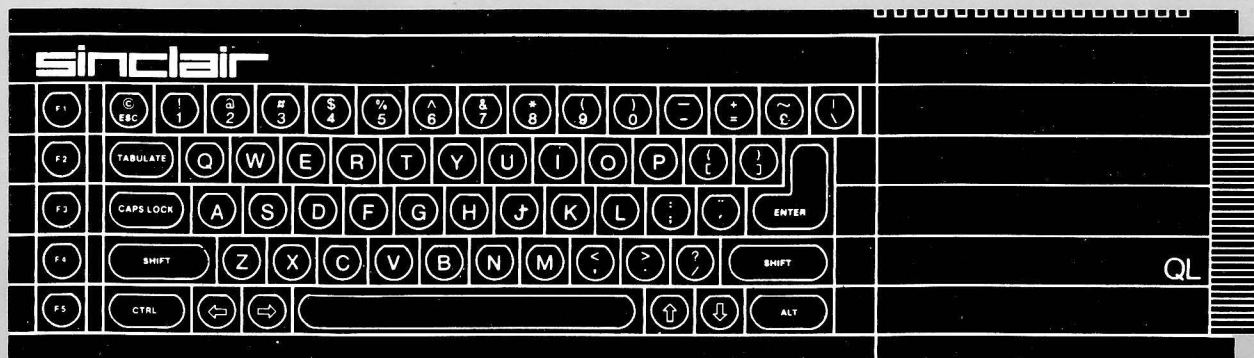
The Sinclair Relocatable Object File format (S-ROFF) — a standard format for relocatable modules — has also been specified. All relocatable compilers and assemblers available through Sinclair will support this format. In order to support the S-ROFF standard and help software houses improve compatibility between compilers, Sinclair Research has commissioned QL Linker from GST Computer Systems. This important new package enables programs written in compiled languages and M68000 assembler to be linked together at execution time, provided they support S-ROFF.

The S-ROFF specification is available from The Software Dept., Sinclair Research Ltd., 25 Willis Rd., Cambridge CB1 2AQ, the linker from GST Computer Systems

Ltd., 8 Green St., Willingham, Cambridge CB4 5JA.

Finally, a full-screen editor written by Metacomco has been chosen by Sinclair to provide a common user interface. It will be included with any compiler or assembler published by Sinclair and can be purchased as a separate product or directly from Metacomco.

Commenting on this initiative, Alison Maguire, our Software Manager, says 'we want to encourage manufacturers of software and peripherals to make use of the facilities for the QL. Everything which is produced for the QL should be designed for compatibility — something we were not able to achieve with the ZX Spectrum. We are determined to make life as easy as possible for QL users'.



SPECIAL OFFER OFFICE

THE Special Offer Office keeps track of all deals and discounts for QJUB members — which includes anybody who is being sent this Newsletter. Listed here are the deals offered to QJUB Members.

Personal Computer News, a leading weekly national computer magazine, offers £5 off the £20 regular price of a year's subscription to PCN. To get in on this deal, write to the PCN Subscriptions Department at the address below, enclosing your £15 cheque or money order and a note detailing your QJUB membership number (if you know it), your name and address. The address to write to is: QJUB Special Offer, Subscriptions Department, Personal Computer News, VNU Business Publica-

tions, 53-55 Frith St., London W1A 2HG.

GST Computer Systems is offering its acclaimed 68K/OS Operating System for the QL at an attractive discount. For details, write to GST Computer Systems Ltd., 8 Green St., Willingham, Cambridge CB4 5JA or 'phone (0954) 81991 and ask for Jane Pateman. Have your membership number and address to hand.

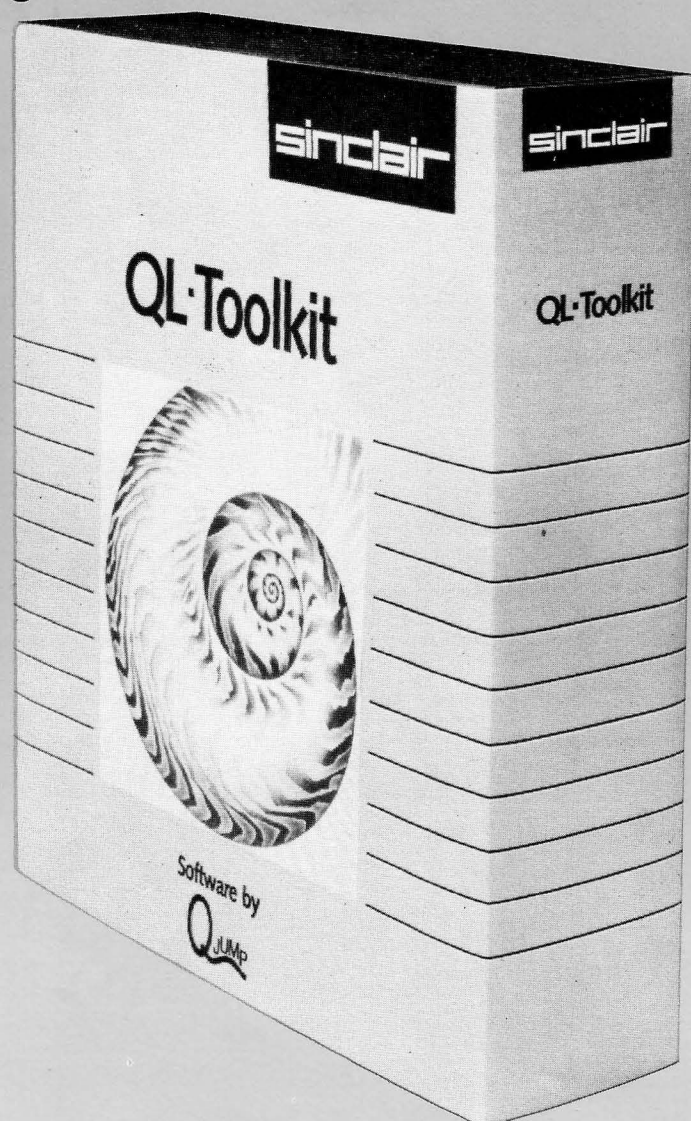
A 20 per cent discount on Sinclair Research Software sold through QJUB. This includes:

QL Chess — QJUB Price £14.95
 QL Assembler — QJUB Price £31.95
 QL Toolkit — QJUB Price £19.95
 QL Cash Trader — QJUB Price £54.95

Send your order or any inquiries about the full list of discounted software to Sinclair Research, Stanhope Road, Camberley, Surrey or 'phone on 0276 685311.

A discount is available for a year's subscription to EMAP's QL User Magazine. A full year's subscription including delivery would normally cost £15. However, if you subscribe through QJUB, the magazine will be delivered for an all-in (postage included) price of £11.50. To receive your 12 issues send a cheque or money order for £11.50 to PRQL Subscription Department, QL User, Priory Court, 30/32 Farringdon Lane, London EC1).

QL-TOOLKIT UNLOCKED



THE long-awaited Tony Tebby Toolkit for the QL is now available.

It's a programmer's toolkit consisting of about 60 programs on Microdrive cartridge, the majority of which can be linked into SuperBASIC from the beginning and then can be used as command extensions or within a program.

Among these SuperBASIC enhancements are: printer spooling — to allow you to print a file while you run a SuperBASIC program, file access — offering full random input/output commands, job control — allowing management of multi-tasking programs including the ability to display, alter priorities and delete multi-tasking jobs from SuperBASIC, an enhanced user interface — provides a full-screen editor, defaulted drive names, the amount of free memory displayed and an extra SuperBASIC command display, fitters and pipe — a unix-like redirectable file input/output system which can generally be accessed through SuperBASIC and a general set of utilities including user-defined graphics, and on-screen real-time clock and format conversions.

The toolkit sells for £24.95-£19.95 to QJUB members — and should have a wide range of applications for a variety of QJUB members.

Next Issue

- A continuing look at sound
- Our much-delayed feature on monitors (promise.)
- Books from Collins
- More on Disk Drives
- And the usual crop of letters, features, stories and jokes.

PROGRAMMER'S FORUM

WELCOME to the QLUB programming forum, where we answer your programming questions and you solve those of other QLUB members.

Question 1:

I HAVE a QL computer and would be grateful for some assistance in connecting a printer to the machine.

The printer I have is a Tandy Model DMP-120 which has a serial input with 4 pin DIN plug which should be connected as follows:-

| Pin No. | Signal |
|---------|----------|
| 1 | Not used |
| 2 | Busy |
| 3 | Gnd |
| 4 | Data |

I have a Sinclair serial printer lead which has 6 wires coloured orange, red, blue, green, white and black.

Could you please advise me as to which wires to connect to the DIN plug and also whether SER1 or SER2 should be used.

E. T. Allan,
Colchester.

ANSWER: In order to answer your question, I'd need to get clearer details on the serial input port than you've given me. Every serial interface consists of a minimum of three 'lines' of information; data in, data out and ground. I gather from your letter that the grounding line is on Pin 3, but that still doesn't tell me where the transmitting and receiving data goes.

However, a little deduction might take us a long way in this case. The QL's grounding wire is black (Pin 7 on the 25-pin Sinclair D-plug). If you're using serial port 1 (SER1), then the input line is white and the output is green.

So for starters, you can hook up the black wire to Pin 3 of your plug and PROBABLY the green wire to your Pin 4 or 'data' line. Whether or not you then go ahead and hook up the white wire (the data input line INTO the QL) to Pin 2 is a moot point. Try it first without.

The reason behind this is that your printer usually only NEEDS to take information in, and not send any back out. So if — as your wiring description suggests — there's only one data line, it's got to be for receiving data from the QL.

If it doesn't work with the white wire either hooked or unhooked to Pin 2, then consider hooking up the QL's blue wire (the DTR signal on Pin 20) to it. That should work.

Once you actually start getting something out of your Tandy printer, you'll also have to be careful about line feeds and carriage returns in your documents. Most Tandy printers ASSUME that the computer supplying data to them will NOT be supplying line feeds. So if you don't configure your printer driver in QL Quill, Abacus, Archive or Easel to generate only a $\frac{3}{4}$ CR÷, then you may end up with constant double-spaced text and figures.

Hint 1:

IN QLUB News No. 2 you published a machine code and SuperBASIC routine for a digital clock/calendar. Somewhat simpler, although maybe not serving exactly the same purpose, I am using the following PROCEDURE to produce a digital calendar/clock, for showing the time and date anywhere in a programme or simply to use my screen as a digital calendar/clock when I am not using my QL for any other purpose.

The PROCEDURE runs as follows:-

```
1 REMark CLOCK: CLOCK = cl
2 DEFine PROCEDURE cl(time)
3 LOCAL time
4 REPEAT time
5 PAPER 1: INK 6: OVER 0:
  CSIZE 0,0: CLS
6 FOR i = 1 TO 365*24*60/32
7 AT 19,17: PRINT DATE $
8 PAUSE 60
9 NEXT i
10 END FOR i
11 END REPEAT time
12 END DEFine cl
```

After having entered exact date and time with SDATE, the command cl will run a digital calendar/clock at the bottom right of the current screen.

Some further remarks:

Outside the United Kingdom line 8 should read PAUSE 50.

Line 5 allows the colour of background and screen to be changed and also the size of lettering and numbers in which the calendar/clock will be printed on the screen. If CSIZE is changed, also AT 19,17 in line 7 should be changed to accommodate the larger lettering or e.g. to print them in the centre of the screen. C212E 3,1 and AT 15,10 will give maximum size lettering in the centre of the screen (TV).

I trust that other QL users will find this very simple PROCEDURE useful.

P. Vernoot,
Prootebroek, Holland.

Hint 2:

THANK you for the first full edition of QLUB News. A most enjoyable and informative edition.

The machine code clock program works well and shows how wrong some of the pundits have been when they said that QL's with ROM version AH would not multi task. So much for their opinions. I think it's a great machine.

However you do seem to have made the error of using the data from the QL manual for the RS232 connections. These are incorrect in one respect. The signal DTR is shown as being on pin 4. In fact on the Sinclair leads there is no wire on pin 4. The signal DTR being on pin 20. A fact that could cause some handshaking difficulties with printers and the like, as it did

with my Kara KP810 which will now run at 9600 Baud with the DTR taken to pin 4.

Now a few notes that are not in the book that may be of use to the hackers out there like myself:-

1. To find the version of ROM use PRINT VER\$.

2. Line numbers may be changed under edit to duplicate program lines.

3. Use the Auto command to do a mass edit. Having amended the line do not press the enter key. Use either the up or down arrow keys. This will take you to the next line and so works like a full screen editor.

4. To run a program directly from power-up name it BOOT.

5. Control + F5 acts as a toggle switch to pause listings etc.

6. Please, please do not use I's and O's as variables in programs. It only causes a great deal of confusion when it is printed and there is really no need for it with a computer like the QL. (Sinclair please take note).

7. I have found that a little drop of silicon (WD40) on the key shafts really makes the keyboard like silk.

8. The QL is a great and very powerful machine. Don't let the knockers put you off. Most of them don't know what they're on about and are just knocking for the sake of it.

A.C. Haddock,
Nuneaton.

Hint 3:

I AM unable to write this letter in QL Quill because my QL is away on a ROM refit. In the latest QLUB bulletin you ask us what we want to use the QL for. I have started on a project to transfer a large simulation program in Pharmacokinetics to the QL. The program is in Fortran and we have been running it generally on the main-frame computers of the University of London. It provides a very good basis for undergraduate course-work in Pharmacokinetics.

I have decided to trim the program down and to rewrite it in SuperBASIC and so far everything has gone well. The program involves a lot of numerical integration which seems to go quite rapidly on the QL. The only disadvantage is that the output for the user is in the form of tables of numbers and I have not been able to set up nice-looking tables. Is it possible to set up tables in QL SuperBASIC similar to those in Fortran? If so I would like to know how to do it.

I enclose an entry for QL Quark challenger of the month.

QL = Quintessential Lorimer.

(Apologies. It may need the Concise Oxford Dictionary to translate it.)

Leonard Saunders,
Loudwater, Bucks.

ANSWER: Any FORTRAN buffs out there who can help?

Music micro please!

THE QL's sound facilities are best suited to making beeps, buzzes and growls to be used in programs serving some useful purpose, such as a 'whooshing' sound when saving Superman from the red Krypton, or sounding an alarm tone for your next appointment. However, if you have bought a QL for its numerous other hardware advantages it seems a shame to miss out on music entirely. The subject is lightly treated in most available text books and in this article we will try to provide some useful information for all you budding musicians.

All sound on the QL is controlled, at least in SuperBASIC, by the BEEP command and careful readers of the QL User Guide will know that BEEP is followed by eight parameters i.e. duration, pitch_1, pitch_2, grad_x, grad_y, wraps, fuzzy and random. Rather understandably, the guide neatly avoids detailed explanation and states that the BEEP command is best used experimentally rather than syntactically. Fortunately, where music is concerned, only duration and pitch_1 are of interest.

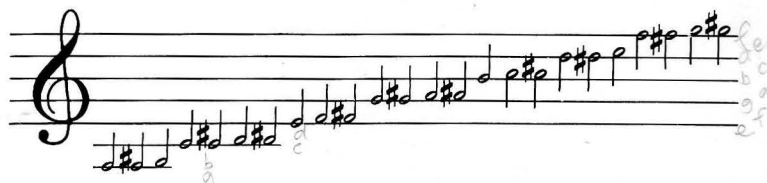
A single musical note will have a pitch, duration and timbre. The timbre is the quality of the note that makes a continuous note of a certain pitch on a trumpet sound very different from the same pitch on a violin. To be more specific, it is the characteristic mixture of the fundamental note and its overtones or harmonics. On the QL any simple attempt to introduce harmonics with the BEEP command is very difficult, so for the rest of this article we will assume simple 'QL timbre'!

The pitch is rather more amenable to control using BEEP and at Fig. 1 is a conversion between musical notes and the pitch_1 parameter of the BEEP command. It has to be admitted that, particularly in the higher notes, the conversion is sometimes inexact but unless you are blessed with perfect pitch the notes can be used to make quite acceptable music. The conversions listed assume that the duration parameter specified to the BEEP command is non-zero; a zero value has a distorting effect, try this program to see:

```
10 REPEAT loop
20 BEEP 0,10
30 PAUSE 20
40 BEEP-1,10
50 PAUSE 20
60 END REPEAT loop
```

If you get a continuous tone you have no problems — otherwise watch out for duration zero!

Having decided that most musical notes over a range of about two octaves can be reproduced, the remaining problem is duration. At first sight there are two



ways of controlling the length of the note. The first is to use the duration parameter within the BEEP command but this is rather curious in action (unless you consider how the 68008 chip communicates with the slave processor) in that durations, allegedly in units of 72 microseconds, go up to 32767 and then start again at -32768. Furthermore, once the main processor has instructed the slave (8049) to make the noise it will itself carry on with the program, which is quite likely to contain another BEEP command. This will also be passed to the slave and will replace the previous note in the 8049's affections, probably long before the sound had finished.

A better method of specifying duration for music is to BEEP a long duration, say -1, to PAUSE a certain number of fiftieths of a second and then to BEEP again using no parameters, thus stopping the sound.

That accounts for the two methods mentioned, although there is a third, which is to BEEP the required pitch, PAUSE the desired time and then to continue with the program which will presumably have another BEEP command. The new pitch will be introduced with no break in the sound, or in musical terms, the notes will be 'slurred' together, a style of playing known as legato. Another style is staccato (and staccatissimo) where successive notes are separated by the full time value but are only sounded for a proportion of the time. This can be arranged by PAUSing say one tenth of the total time value, sounding the note for eight tenths and then PAUSing a further tenth. See Fig. 2.

The final problem we have with duration is to decide how long each note of a given musical value is to last. Beethoven often helped out with a metronome mark which would indicate the number of crotchet beats to the minute, say 160. Fig. 3 gives the musical representation of the various notes and their time values related to a crotchet.

Enough theory. The program listing is guaranteed to annoy the rest of the family and send the cat skulking to a corner. It asks if you would like it to make some music (auto) or wish to enter some yourself (manual) and then draws a musical stave complete with treble clef. The notes appear and are sounded in shortened form as they are selected; after eighteen notes you may play the resulting 'tune', complete with a marker for those who cannot read music. Because the duration of the notes is controlled by a PAUSE command the notes may be cut short by hitting the 'any' key — you may therefore impose your own rhythm on the tune by judiciously tapping a key.

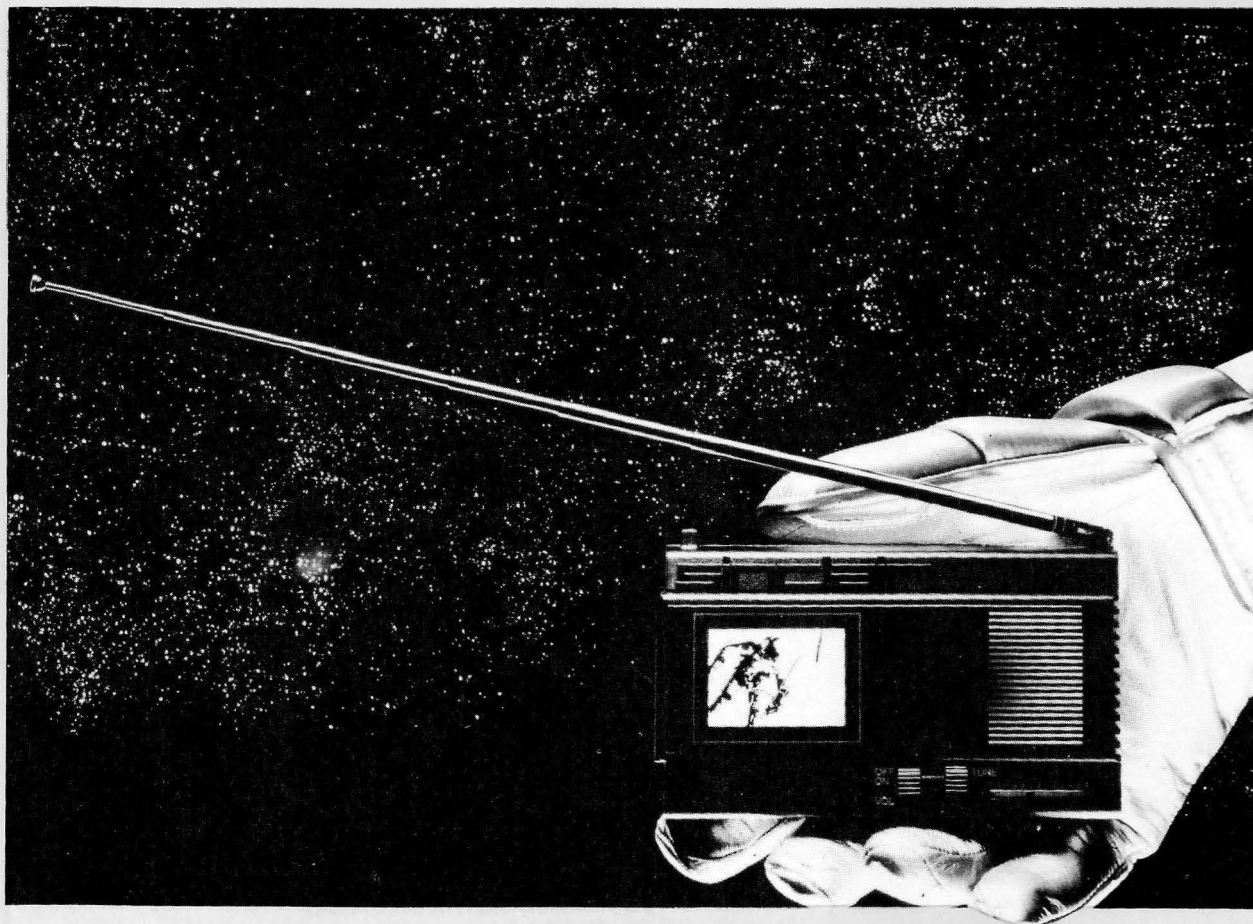
The program is not a serious attempt to turn the QL into a musical instrument, more a bit of fun, but it has been written in a very structured way (no GOTOS!) and it would be quite easy to add user inputs for duration as well as pitch, store more notes over a wider range and include simple editing facilities. Have fun!

```

100 DIM pitch(18)
110 MODE 4:WINDOW 448,200,32,16
120 WINDOW 0, 448,20,32,216
130 PAPER 4:PAPER0,7:INK0,0
140 CLS0:PRINT0,"Auto or manual ? (a/m)"
150 IF INKEY$(-1)=="m" THEN yourself=1:ELSE yourself=0
160 REPEAT loop
170 up=50:across=16:inc=0
180 CLS
190 STAVE
200 DRAW_CLEF
210 FOR note=1 TO 18
220   PICK_NOTE yourself
230   DISPLAY_NOTE
240   pitch(note)=p
250   across=across+8
260 END FOR note
270 PLAY_TUNE
280 CLS0:PRINT0," Another tune ? (y/n) "
290 again$=INKEY$(-1)
300 IF again$=="n" THEN EXIT loop
310 END REPEAT loop
320 STOP
330 DEFINE PROCEDURE PICK_NOTE(yourself)
340 IF yourself
350   REPEAT check
360   CLS0:INPUT 0,"Note number ? (1 to 9) ";choice
370 ELSE
380   choice=RND(1 TO 9)
390 END IF
400 SELECT ON choice
410   =1:p=24:inc=0
420   =2:p=22:inc=1.5
430   =3:p=19:inc=3
440   =4:p=15:inc=4.5
450   =5:p=12:inc=6
460   =6:p=11:inc=7.5
470   =7:p=9 :inc=9
480   =8:p=7 :inc=10.5
490   =9:p=6 :inc=12
500   =REMAINDER :END REPEAT check
510 END SELECT
520 END DEFINE
530 DEFINE PROCEDURE DISPLAY_NOTE
540 FILL 1
550 CIRCLE across,up+inc,1.5
560 FILL 0
570 IF p < 12
580   LINE across-1.5,up+inc TO across-1.5,up+inc-8
590 ELSE
600   LINE across+1.5,up+inc TO across+1.5,up+inc-8
610 END IF
620 BEEP -1,p:PAUSE 10:BEEP
630 END DEFINE
640 DEFINE PROCEDURE STAVE
650 INK 7
660 FOR ledger=0 TO 12 STEP 3
670   LINE 2,up+ledger TO 165,up+ledger
680 END FOR ledger
690 INK 0
700 END DEFINE
710 DEFINE PROCEDURE DRAW_CLEF
720 LINE 8,up+1.5
730 ARC_R TO 0,4.5,-PI
740 ARC_R TO 0,-6,-PI TO -3,7,-3*PI/4
750 LINE_R TO 5,7:ARC_R TO -2,0,PI
760 LINE_R TO 0,-18
770 FILL 1:CIRCLE_R -1,0,1:FILL 0
780 END DEFINE
790 DEFINE PROCEDURE PLAY_TUNE
800 CLS0:PRINT0," Press any key to play: "
810 PAUSE
820 x=16:y=48
830 FOR note=1 TO 18
840   BLIP x,y
850   BEEP -1,pitch(note)
860   PAUSE 20
870   BLIP x,y:x=note*8+16
880   BEEP
890 END FOR note
900 CLS0:PRINT0," Play again ? (y/n) "
910 IF INKEY$(-1)=="y":PLAY_TUNE
920 END DEFINE
930 DEFINE PROCEDURE BLIP (x,y)
940 INK 4
950 OVER -1
960 CURSOR x,y,0,0
970 PRINT "*"
980 OVER 0
990 END DEFINE

```

Your chance to win one of 3 new Sinclair pocket TVs.



THE QLUB News competitions desk is awash with entries from the first two competitions, and you'll see below the details and pictures of the winners of the first competition receiving their QL Vision colour monitors at the awards ceremony on January 16th in London. We'll print the names of the second competition winners next issue, although we will be sending out their prizes long before then.

THIS competition gives you a chance at a prize which will be of more use to you than to your QL — a revolutionary new Sinclair flat-screen pocket TV. We're offering three TVs and all you have to do to get your chance at winning one of them is answer all the questions below correctly. The first three correct entries we draw will win. Entries must be in by no later than April 15, 1985 to qualify. Winners' names will be published in the July/August edition of QLUB News and prizes will be awarded in May.

Here are the questions:

- 1) How many silicon chips are used in the Sinclair flat-screen pocket TV?
- 2) When was the first primitive colour

television transmission demonstrated?

a) 1925; b) 1929; c) 1940; d) 1955.

3) Which British inventor is often credited with inventing the first mechanical system of television?

When you think you've got the answers, write them on a piece of paper along with your QLUB membership number (just write 'new member' if you've haven't been assigned a number yet) and send the entry to:

QLUB Competition

The Editor

QLUB News

Sinclair Research Ltd.

25 Willis Road

Cambridge CB1 2AQ



PICTURED above: Sinclair Research's Nigel Searle holding up his end of the prize-giving as he hands over QL Vision colour monitors to competition winners Mr J. Mounsey of Nottingham, Mr P. Brown of East Barnet and Mr J. Wilkinson of Chesterfield at the January 16th ceremony in London. A slap-up meal was awarded to the editor and prize-winners after the presentation. The editor was said to be appropriately grateful and keen for more competitions.

The Psion problem page.

PROBLEM: What to do when you find QL ARCHIVE fields have scrambled text/numbers.

ALTHOUGH you can have fields up to 255 characters in length, when you display them on the screen you must either string slice the fields or take care to keep your insertions to the right length so that they fit into the space you have allocated for them. Data fields will not be corrupted by this occurrence, i.e. if lines\$ is a text field of over 80 characters then LET part1\$=line\$ (to 80): LET part2\$=line\$ (81 to 160) will slice line\$ into parts which can be incorporated in your SEDIT screen design. This should be done inside a simple procedure.

```
proc p
LET part1$=line$ (to 80): LET
part2$=line$ (81 to 160)
endproc
```

Alternatively allow space in your screen design to superimpose the field by using PRINT AT from inside a procedure.

```
proc p
print at 6,0; lines$
endproc
```

Or use more than one field to hold the text i.e. line1\$, line2\$ and keep insertion to the fields to the length of space available in your screen design.

AUTHORS :

```
auth1$
auth2$
```

TITLE :

```
title1$
title2$
```

SOURCE :

```
source1$
source2$
```

PROBLEM: QL Archive appears to allow only 160 characters per field.

QL ARCHIVE will allow 160 characters to be added to a field using INSERT or ALTER. If larger fields are required, up to the maximum of 255 characters, then additional characters can be added to the field. For example:

LET comment\$=comment\$+B\$ will add the contents of B\$ to comment\$. Be careful to check that comment\$ will not exceed 255 before adding B\$ LEN(). Use PRINT AT in your procedure to position comment\$ on the screen. An easy way to handle text in QL ARCHIVE (and the method used for this letter) is to use a field for each line of text.

PROBLEM: Using EPSON FX80 compatible printers with QL EASEL.

THE printer driver for the EPSON FX80 is set to a baud rate of 9600. If your printer is set to a different baud rate, and you cannot change it, then you can modify the 'BOOT' program instead. To do this, type in the following with a cartridge containing QL EASEL in MDV1:—

```
LOAD "MDV1_BOOT"
10 BAUD baud_rate (type in the baud
here)
DELETE "MDV1_BOOT"
SAVE "MDV1_BOOT"
```

If you use a slower baud rate, then it is best to RUN the following SuperBASIC program as well:—

This program allows you to alter the timeout (the amount of time QL EASEL will wait before it decides to stop sending data to a printer, slow printers will cause QL EASEL to timeout), changing the end of line code to carriage return as opposed to carriage return and line feed or change the device to which QL EASEL sends the data.

```
10 LET tout=3000
20 LET a=respr (3500)
30 LBYTES mdv1_gprint_prt,a
35 REM to alter timeout
40 POKE_W a+94,tout
45 REM to output CR only
50 POKE a+109,1
55 REM to change Device type (change
line 60 to suit)
60 device$="par1"
70 delen=len(device$)
80 if delen<6 then print "Device name
too long": goto 200
90 POKE_W a+184,delen
100 FOR i=1 to delen
110 POKE a 185+i,CODE(device$(i))
120 NEXT i
125 REM Now Save File
200 DELETE mdv1_gprint_prt
210 SBYTES mdv1_gprint_prt,a,2000
```

Before you run the program, make sure that you have a CLONEd copy of QL EASEL in MDV1.

Note: this is only applicable to QL EASEL version 1.01. Later versions will not need to be changed in this way (although the baud rate may still need to be set).

PROBLEM: Printing colour graphics from QL EASEL

EARLY versions of QL EASEL only contained a graphics printer driver for a FX80. However this driver does, or with simple modifications, work with most EPSON FX80 compatible printers.

Psion is presently writing printer drivers for some of the more popular printers used by QLUB members. The first colour printer driver to be produced to work for QL EASEL will be the INTEGRIX/CANNON INK JET and a JX80. There will also be a printer driver for a pen plotter. As a QLUB member, you will be notified of any developments in this field as they occur.

PROBLEM: QL QUILL only supports BOLD, UNDERLINE, SUBSCRIPT, SUPERScript

QL QUILL will only support the typefaces that you can display on the screen (i.e. Bold, Underline, Subscript, Superscript). If you wish to print in a different typeface you must either use the preamble/postamble options in INSTALL_BAS to set up the printer in a particular typeface or use one of the existing typeface selections to send your required printer control codes. If, for example, your printer will not print subscripts then you could use INSTALL_BAS to place in the subscript line the codes needed to change to a condensed typeface.

The Translate options in INSTALL_BAS can also be used to convert characters in your document to control codes which select different typefaces.

PROBLEM: Using the QL ABACUS UNIT command and its effect on the displayed numbers.

USING UNITs to change the way that numbers are shown on the screen does not convert any numbers. For example use UNITs to change defaults to decimal with 3 decimal places. Enter 2.005 into a cell and you will see 2.005 displayed in the cell and at the bottom left of the screen. Now change to two places of decimal. Notice that the 2.005 has changed to 2.01 but 2.005 is still displayed in the bottom left of the screen. QL ABACUS uses the true values in calculations: in this case it will use 2.005.

Machine Code and More

In every pursuit there are people who seek out information at a level not generally pursued by others – specialists who work hard to be as enthusiastic and well-informed as possible. This page, Machine-code and more, is for just those people. While other sections of QLUB News will deal with SuperBASIC programming and questions of more general interest, this page will be reserved for those with a keen eye for – and an interest in – machine code programming. The QL will indeed be a challenge for you, but by sharing information we can help you meet that challenge.

WHEN the QL communicates with the outside world, it does so by passing information down streams through channels which are connected to devices. When you type a statement like

```
OPEN#3,mdv1_afil
```

you are connecting the device mdv1_afil to channel number 3. As Microdrives are media with directories, you could split the device definition further into the physical device – mdv1 – and the file – afile.

When you print things on the screen or read from the keyboard, we are still using channels and devices. There are two devices for screen communication: the SCREEN device and the CONSOLE device. A screen device is opened using the scr device name, as in

```
OPEN#3,scr_512x244a0x12
```

and you can only send output to it; you can't read input from it. The console device, on the other hand, is capable of both input and output, because it adopts the screen for output and the keyboard for input. As you can often see what you're inputting, the console is often also described as 'interactive'. The name used by Qdos to signify a console device is con, so you can open a channel to a console device:

```
OPEN#3con_512x20a0x0_12B
```

When you open a console or screen device, you specify the size and position of the window on the screen which the device will use. In our screen example above, we declared that the window would be 512 pixels wide, 244 pixels high and at position (0,12). Whenever textual output is directed to a screen window, it is placed at the current position of the cursor. Normally, you can only see the cursor on channel 0, the command channel, but by resorting to machine code you can switch the cursor of each channel on and off at will.

When a cursor attached to a console device awaiting input is on, or enabled, you can switch between that channel and others awaiting input by pressing CTRL and C simultaneously. As it is fairly usual for independent multi-tasking programs to have no more than one channel awaiting input at a time, you'll find this facility useful for 'changing jobs'. By selecting a particular cursor with CTRL-C, you select the job owning that cursor's channel.

Apart from specifying the size and position of a window, you also decide on

its paper, ink and strip colours, and the size and colour of its border. Opening channels and windows is almost as easy from machine codes as it is from SuperBASIC, as there are two routines inside called UT.SCR and UT.CON to do it for you. These routines take exactly the same parameters, but the first opens a screen device while the second opens a console device. Below is a typical invocation of one of these routines.

```
OPEN_WIN MOVE,W UT.SCR,A2
LEA.L PBLOCK,A1
JSR (A2)
```

...rest of program...

```
PBLOCK DC.W $701 Border width 1,
          colour 7 (white)
DC.W 2    red ink (2) on
          black paper (0)
DC.W 512  window width
DC.W 244  window height
DC.W 0    window X-origin
DC.W 12   window Y-origin
```

Both utilities are vectored, which means that their addresses are held in a specified memory location. By loading this address into an address register and jumping to the subroutine there, you'll find it particularly simple to invoke each routine. UT.SCR and UT.CON both need register A1 to point to a parameter block, which defines the window. Ours is at PBLOCK. This block is made up in the following way:

```
Byte 0:    the border colour
Byte 1:    the border width
Byte 2:    the paper and strip
           colour
Byte 3:    the ink colour
Bytes 4 and 5: the width
Bytes 6 and 7: the height
Bytes 8 and 9: the X-origin
Bytes 10 and 11: the Y-origin
```

The routines return with either the channel ID in register A0 and the zero flag set, or an error code in D0 and the zero flag reset. Errors can range from 'Out of range', which means that the window parameters were a bit silly, to 'Out of memory', which means that there isn't enough space inside the machine to open a new channel.

The channel ID is a long word (32 bits) which must be preserved for as long as a channel needs to be used, as it uniquely identifies that channel to Qdos. If you were to look at the channel ID closely, you would see that it is in two parts – a word (16 bits) which is used to 'index' into the channel table which Qdos

maintains, and a further word which is called the 'tag', which is used to separate a channel from a no longer existent one which occupied the same slot in the channel table. Neither bears any relation to the # channel numbers used in SuperBASIC, and the channel ID as a whole is only valid while the channel is open.

Once we have a screen or console channel's ID, and assuming that the channel has not subsequently been closed, you can use any of the routines inside Qdos to send information to it, read information from it and perform miscellaneous operations such as panning, scrolling and recolouring. Most of these are done using one of the 68008's trap exceptions, TRAP #3 to be precise. In all these routines, and every other which deals with channels, the channel ID must be passed to register A0.

There are another couple of vectored routines which allow us to print messages, numbers and errors to a specified channel. These are UT.MTEXT, UT.MINT and UT.ERR respectively. They are called in much the same way as UT.SCR and UT.CON, but obviously need different parameters. UT.MTEXT needs the address of the string to be printed in register A1. This string must be in normal Qdos format, which means that the characters of the string must be preceded by a word containing the length of the string, and A1 must point to this. UT.MINT requires the integer (a whole number between -32768 and 32767) to be printed in register D1, while UT.ERR needs the Qdos error code (-1 to -21) in register D0. Naturally, they all need the relevant channel ID in register A0.

All the screen handling routines which are invoked by TRAP #3 follow a fairly standard format, in which register A0 holds the channel ID, D3 holds the timeout and registers D1, D2 and A1 hold any parameters which are required. Register D0 is used to tell Qdos which particular routine is being accessed. The timeout is a number between -1 and 32767 which determines the number of 50ms intervals for which the routine will wait until its desired effect is achieved. In the case of a timeout of -1, it will wait forever. All routines return with either 0 or an error code in D0, and if the routine cannot finish within the timeout period, the error will be 'not complete'.

To demonstrate window handling and

the use of some of these routines, we'll write a short program to copy a named file to the screen. We'll write it in such a way that it will run as an independant job, or multi-tasking program. It uses a lot of the vectored utilities and TRAP #3 routines we've discussed here.

The code starting at the BEGIN label opens a console device as previously described, and saves the channel ID on the stack. It then uses UT.MTEXT to print a message saying 'Filename: ' to this console window, and collects a line of input from the keyboard using the very useful IO.FLINE trap routine. This collects bytes from the specified channel until either more bytes than expected are collected or a newline character is met. The maximum number of characters to read is specified in D2, and each character is put into the buffer pointed to by A1. This routine is particularly useful as it allows the user to edit the line being entered in the normal QL way. When the routine returns, register D1 holds the number of characters read including the terminating

newline. By subtracting 1 from D1 we are left with the number of real characters read in. Putting this in the word preceding the first character gives us a Qdos format string, which we pass to the IO.OPEN routine. This decodes the string as if it were a device name, and then attempts to open a channel connected to that device. We specify how that channel should be opened by the value of D3. If, as in this case, it contains 1, the channel is opened in such a way that it can only be read.

If this channel, is opened successfully, you save the ID on the stack and proceed. If not, you must print out an error message to the console and try the whole process all over again. You use the UT.ERR utility to do this. When you have finally opened your file, you enter a loop in which a byte is read from the file, using IO.FBYTE, and then transferred to the console using IO.SBYTE. If an error condition occurs during the IO.FBYTE call, then it either means that you have reached the end of the file, or some unexpected error, such as

'bad medium', has occurred. To find out which, we compare the value of D0 against that of the 'end of file' error. If this was the error condition, we end the program, but if another error has occurred we report it to the console beforehand.

The closing sequence simply uses IO.CLOSE to close the two channels, and MT.FRJOB to kill the job. At this point, it is worth noting that no code needs to follow the TRAP #1 instruction there, as that call will remove the job from memory — the 68008's program counter will not point to this location again until a new job is loaded. Some authors in some magazines have been putting entirely redundant (and semantically meaningless) RTS instructions here.

With the host of routines available within Qdos, it is a very simple matter for even novice machine code programmers to play around with window and screen handling. In later issues we'll be looking at methods of implementing high resolution graphics from machine code.

* A program to copy a named file to a screen device - for QLUB by Adam Denning
* 30th January 1985

SIZE 100 Use 'DATA' for Sinclair Assembler
100 bytes is enough for the buffer and
the job's A7 stack

* Equates, constants, and so on - these would normally be in a header file

MT.FRJOB EQU 5 trap 1 key for force job remove
IO.OPEN EQU 1 trap 2 key for general channel open
IO.CLOSE EQU 2 trap 2 key for channel close
IO.FBYTE EQU 1 trap 3 key for single byte read
IO.FLINE EQU 2 trap 3 key for LF-terminated line read
IO.SBYTE EQU 5 trap 3 key for single byte write
UT.CON EQU %C6 vector address for console open utility
UT.ERR EQU %CC vector address for error printing utility
UT.MTEXT EQU %DD vector address for message print utility
OPEN_INS EQU 1 D3 key for IO.OPEN - shared read access
ERR.EF EQU -10 error code for 'end of file'

* The beginning of the code

START BRA.S BEGIN jump to start of program
DC.L 0
DC.W %A4FB 'standard format' identification
DC.W 4 program name - length first
DC.B 'QLUB' then the actual characters

* A window definition block for UT.CON

PBLOCK DC.B 7 border colour (white)
DC.B 1 border width
DC.B 2 paper colour (red)
DC.B 7 ink colour (white)
DC.W 512 window width - 512 pixels
DC.W 100 window height - 100 pixels
DC.W 0 window origin at (0,0)
DC.W 0

* Yes, the program starts here!

BEGIN LEA.L PBLOCK,A1 Open the defined window
MOVE.W UT.CON,A2 as a console device
JSR (A2) ...we assume no errors
MOVE.L A0,-(A7) now save channel ID

GETNAME LEA.L FILENESS,A1 print the FILENESS prompt
MOVE.W UT.MTEXT,A2
JSR (A2)

LEA.L BUFFER+2,A1 put buffer (+2) address in A1
MOVEQ #1,D3 signal infinite timeout
MOVEQ #IO.FLINE,D0 read a string into the buffer
MOVEQ #80,D2 up to 80 characters
TRAP #3 do it

* We assume no errors from IO.FLINE call

LEA.L BUFFER,A0 get address of buffer in A0
SUBQ.W #1,D1 'erase' terminating line feed
MOVE.W D1,(A0) store string length at start of string
MOVEQ #IO.OPEN,D0 get ready to open file/device
MOVEQ #1,D1 for this job
MOVEQ #OPEN_INS,D3 with shared read access
TRAP #2 now do it
TST.L D0 any errors?
BEQ.S GOTFILE branch if not
MOVE.W UT.ERR,A2 print QDOS error message
MOVE.L (A7),A0 to our console channel
JSR (A2)
BRA.S GETNAME and then try it all over again
GOTFILE MOVE.L A0,-(A7) save the file channel ID
MOVEQ #1,D3 signal infinite timeout - D3 is preserved
by both IO.FBYTE and IO.SBYTE below

* Now transfer the file byte-by-byte to the screen. By using IO.FSTRG and
* IO.SSTRG instead of IO.FBYTE and IO.SBYTE, we could make the transfer more
* efficient, but we won't!

COPYFILE MOVE.L (A7),A0 put file channel ID in A0
MOVEQ #IO.FBYTE,D0 prepare to read a byte from it
TRAP #3 do it
TST.L D0 any errors?
BNE.S LASTCOPY yes, so branch

* Byte read is now in D1.5

MOVE.L 4(A7),A0 put console channel ID in A0
MOVEQ #IO.SBYTE,D0 prepare to send a byte to it
TRAP #3 do it
BRA.S COPYFILE assume no errors and continue

* There was an error during IO.FBYTE. If this error was 'end of file', then it's
* OK and means we've finished. Otherwise we'd better print an error message first

LASTCOPY CMP1.W #ERR.EF,D0 'end of file' error?
BEQ.S KILLIT yep!
MOVE.L 4(A7),A0 else get console channel ID
MOVE.W UT.ERR,A2 and print error message
JSR (A2)

KILLIT MOVE.L (A7)+,A0 get file channel ID
MOVEQ #IO.CLOSE,D0 and close it
TRAP #2

MOVE.L (A7)+,A0 get console channel ID
MOVEQ #IO.CLOSE,D0 and close it
TRAP #2

MOVEQ #MT.FRJOB,D0 now force kill
MOVEQ #1,D1 this job
MOVEQ #0,D3 with no return error (if EXEC_W-invoked)
TRAP #1 the end!

FILENESS DC.W 10 prompt message
DC.B 'Filename.'

BUFFER EQU * the job's data area starts here
END

QL Quarks

THE response to both our QL Quark challenges was so overwhelming, we've only just picked the staff of QLUB News off the floor from all the mirth and laughter they caused. You'll recall that we set two challenges; first to come up with an alternative to what QL stands for and the second to complete the limerick which began:

A QL computer from Cambridge ...

Below we present some of the more entertaining (and printable) responses to the challenges.

The young Dalai Lama
Used to live in Tibet
But the Chinese invaded
Now his house is to let.

From the roof of the world
And the rarefied air
He wandered the earth
'Til he met Clive Sinclair

"How QL thy fate
Aba cuss, aba bun
If you settle with me
I will make you Psion."

With the QL adopted
The software ea sel,
"Ar chive" he would murmur
"Quill death us do quell."

C. R. Adams,
Grindleford, Derbyshire.

Ignoring those alternatives which have already been published (i.e. Quantum Lope, Quite Late, Quantity Limited, etc.)

How about:

Quite Ludicrous — The way the machine and its bundled software are sometimes criticised when being compared to systems costing several times as much.

Quirkishly Luscious
(Quirky Lusciousness) — Never mind industry standards, this'll sell on looks alone. Normally-scanning 80-column displays and centronics interfaces have got nothing on black Scandinavian ergonomically designed keyboards.

Joking and cynicism aside, it has to be said that the QL is Quaintly Lovely and Quite Laudable.

Tony Rawcliffe,
Leyland, Lancs.

QL stands for "Quintessential Linchpin."

M. Gibson,
The Sands, Durham.

QL = Queer Leads

D. Willey,
Scarborough.

QL Quarks. My AI program gives the following selection of names for the machine.

Quasi Lobotomy
QUANTUM LIBET
Peculiar Lysid
Quince Limewort Tie
Pai Lateral
QUIXOTIC LUXURY
Quintilly Laughable
QUALUCIDUS
Quid Regatum
Quintessential Lye
QUARRELOUS LAMENTATION
Quibble Lever

E. J. Sargeant,
Long Eaton, Nottingham.

To be consistent with the original Sinclair advertising documentation of the QL showing the user manual with sections written in Latin, I suggest

"Quod Libet"

as a humorous alternative to Quantum Leap.

Gerard M. Delaney,
London, W11.

QL = Quick Loading?

P. G. Mallett,
Scarborough, N. Yorks.

QL stands for Qompter Lure, Quarry Lust, Quite Lucky (if you didn't get a dongled version) or Quite Late.

David Johnson,
Balsall Common, West Midlands.

A QL computer from Cambridge
Had a program packed with Quark language

Quite quick with his quips
Clive soon got to grips
Qeverly converting Quarks to Qlives
QLanguage.

E. J. Davey,
Hereford.

A QL computer from Cambridge
Went south to play Chess in Weybridge

"It just isn't fair,
The 'Knight' is Sinclair"
And so decided to play Bridge

(Name not given)
South East London College,
Lewisham Way, London.

A QL computer from Cambridge
Bit a byte from a Fawltly programme
Which so dinged his dangle
He burst into song
Thus inventing a Qsical longuoge

G. R. Adams,
Grindleford, Sheffield.

Thanks for your issue No. 3 which I have read — every word — even those on page 15 which I did not understand.

I could not help rising to your limerick challenge.

A QL computer from Cambridge
Worked while one munched on a sandwich
We fear at the end
It was no longer a friend
But a crumb crunching quondam —
a damn' fridge.

A poor thing no doubt, but might help fill the corner.

Stanley Craig,
Vicarage Lane, London E6.

A QL computer from Cambridge
Was used by Sir Clive in a game which
With Easel and Quill
As Psionists will
He won all the Qdos and became rich.

Derek Fairburn,
Kenilworth, Warwickshire.

ALTHOUGH long hours were spent pondering over these entries — and the many we didn't have room to print (again, thanks to everyone who entered) — the editor just couldn't stop laughing over the letter below and awarded its author — M. Kingston of Cheshire — the limerick prize of a QL Toolkit program. Here's the winning letter:

A QL computer from Cambridge
Was sold to a family called Fainbridge
But they were such nits
They bashed it to bits
And used the parts to modernise their
old fridge!

M. Kingston,
Poynton, Cheshire.

MEANWHILE, despite the flood of entertaining replies to an alternative for Quantum Leap as the origin of the QL's initials, we managed to pick a winner for the free Microdrive cartridges. The winner is W. T. Cowhig who will receive — not two — but *four* Microdrives plus a Microdrive wallet for this outstanding entry!

I find the keyboard quite agreeable to use after some practice and for this and its other virtues I would describe the machine as:

Quintessentially Lightfingered!

W. T. Cowhig,
Sale, Cheshire.

That's all we've got room for in this issue. But to keep you busy, we'll offer a copy of the 'QL Touch 'n' Go' typing tutor program for the funniest story — in 250 words or less — about you and your QL.